

# Open Research Online

---

The Open University's repository of research publications  
and other research outputs

## An Inquiry Into The Nature Of Analysis.

### Thesis

How to cite:

Brinklow, Anthony (2000). An Inquiry Into The Nature Of Analysis. MPhil thesis The Open University.

For guidance on citations see [FAQs](#).

© 1999 Anthony Brinklow

Version: Version of Record

Link(s) to article on publisher's website:

<http://dx.doi.org/doi:10.21954/ou.ro.0000ff80>

---

Copyright and Moral Rights for the articles on this site are retained by the individual authors and/or other copyright owners. For more information on Open Research Online's data [policy](#) on reuse of materials please consult the policies page.

---

[oro.open.ac.uk](http://oro.open.ac.uk)

# An inquiry into the nature of analysis

by

Anthony Brinklow BSc

A thesis submitted for the degree of

Master of Philosophy

at

The Open University

1999

Disciplines: Software Engineering, Strategic Management, Organisational Behaviour,  
Cognitive Psychology and Complexity Theory

Date of Submission: December 1999

DATE OF SUBMISSION: 1 NOVEMBER 1999

DATE OF AWARD: 27 MARCH 2000

ProQuest Number:27727934

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 27727934

Published by ProQuest LLC (2019). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code  
Microform Edition © ProQuest LLC.

ProQuest LLC.  
789 East Eisenhower Parkway  
P.O. Box 1346  
Ann Arbor, MI 48106 – 1346

## **Abstract**

Increases in the complexity and uncertainty of corporate activity indicate that the time is now appropriate to review the analysis process. This proposition forms the central theme of the thesis, i.e. to explore the nature of analysis.

Initial research concentrated on the field of hard system methods, to provide a theoretical foundation for conducting analysis. However, from observations undertaken as a reflective practitioner it became clear that, even with theoretical advances, hard system methods could only make a marginal contribution to the analysis process. Hard system methods failed to account for the fact that experts had an uncertain knowledge of the domain on which they were expected to pronounce.

Contemporary literature from the fields of strategic management and organisational behaviour pose fundamental challenges to the accepted origin and nature of requirements for change. Complexity theory, however, offers a new theoretical foundation to ease the plight of the domain expert, i.e. pattern recognition. However, to ensure that patterns reflect the cognitive strategies and priorities of the domain expert, it is necessary to explore the field of cognitive psychology to appreciate the significance of the metaphors selected to construct patterns. Finally, knowledge management claims that the value of knowledge is under endless assault and argues for the domain expert to be engaged in a virtuous cycle of perpetual knowledge creation. The thesis seeks to integrate these themes to redefine the analysis process based on methodological pluralism. The key to methodological pluralism proved eventually to be the introduction of generic 'behaviour accentuated' patterns of analysis at the core of the selected techniques.

The nature of analysis has changed radically over the last decade and significant research is required to develop themes raised in this thesis. Moreover, further work is required to disseminate the themes to the practitioner community.



## **Acknowledgements**

I have spent well over two decades working in the IT industry. Experience gained over this period had led to some disquiet about the role of analysis in the industry and this provided the initial inspiration for the thesis. However, embarking on a post-graduate course of study raised the possibility that it was necessary to look beyond IT to discover the fundamental elements of analysis.

To better understand the analysis process, it was necessary to explore the fields of strategic management, organisational behaviour, knowledge management, cognitive psychology and complexity theory.

I would like to extend my profound gratitude to Professor Pat Hall and Professor Roland Kaye for their patient support and encouragement in attempting to forge these disciplines into a coherent process for analysis. Above all, I would like to thank them for imposing the academic rigour and discipline necessary to complete the thesis.

---

# Table of Contents

1	Introduction.....	1
2	The Failure of IT .....	7
2.1	Introduction .....	7
2.2	The Evidence of Failure.....	8
2.3	Other perspectives on IT failure .....	11
2.4	The Elicitation and Expression of Requirements .....	16
2.5	The Source of Requirements .....	21
2.6	The Strategic Planning Paradox .....	25
2.7	The Nature Of Requirements.....	28
2.8	Chaos and Complexity.....	29
2.9	Holism versus reductionism .....	32
2.10	Conclusion.....	34
3	Domain Knowledge and the Learning Organisation.....	37
3.1	Introduction .....	37
3.2	Simple and Complex Organisational Learning.....	39
3.3	Triple Loop Learning.....	44
3.4	The Emergent Nature of Knowledge .....	45
3.4.1	Ross's 'ladder of inference' .....	46
3.4.2	The 'left-hand column' research method .....	47
3.4.3	Senge's approach to organisational learning.....	48
3.4.4	Claxton's hare and tortoise metaphor .....	50
3.5	The Primacy of Tacit Knowledge .....	53
3.6	Nonaka and Takeuchi's Knowledge-Creating Company.....	54
3.7	Requirements - a reprise .....	63
4	The Knowledge Creation Cycle .....	65
4.1	Introduction .....	65

4.2	The Structure of Information.....	66
4.3	Polyani's modes of knowing.....	68
4.4	Kolb's learning typology.....	70
4.5	Requisite variety for abstraction and codification conventions.....	72
4.6	The Transfer of Knowledge.....	72
4.7	Boisot's Information-space (I-space).....	78
4.8	Boisot's Social Learning Cycle (SLC).....	79
4.9	The erosion in the value of knowledge.....	80
4.10	Barriers to the SLC and the liberating influence of entropy.....	82
4.11	Towards a unified view of knowledge creation.....	83
4.12	The Implications for Analysis Methods.....	85
4.13	The role of models in managing complexity.....	88
4.14	Towards a new agenda for analysis.....	88
5	The Limitations Of Analysis Methods.....	91
5.1	Introduction.....	91
5.2	A note on systems.....	91
5.3	The dominant trends in the analysis of systems.....	92
5.4	Structured Analysis.....	96
5.5	Information Engineering.....	100
5.6	Object Orientation.....	108
5.6.1	The Fundamentals of OO.....	109
5.6.2	The Responsibility-Driven Method (RDM).....	111
5.6.3	The Coad & Yourdon method.....	113
5.6.4	The Object Modelling Technique (OMT) method.....	115
5.6.5	The Ptech method.....	119
5.6.6	Object-Oriented Information Engineering (OOIE).....	125
5.6.7	The Object-Oriented Software Engineering (OOSE) method.....	127

5.6.8	The Booch Method .....	134
5.6.9	The Unified Modelling Language (UML) .....	139
5.7	OO methods - an evaluation .....	142
5.8	Analysis methods - a final evaluation .....	148
6	The Role of the Business Process .....	150
6.1	Introduction .....	150
6.2	What is a Business Process? .....	152
6.3	The Value Analysis approach to defining processes .....	156
6.4	The Analysis of Business Processes .....	158
6.5	The Identification and Definition of Business Processes .....	160
6.6	The Behavioural Barrier to Knowledge Creation .....	166
6.7	The Role of Reconstructed Processes in the Knowledge Creation Cycle .....	169
7	The Contribution of Analysis Patterns to Knowledge Creation .....	171
7.1	The Case for Patterns .....	171
7.2	Pattern languages as an extended codification convention .....	173
7.3	The Problems Associated with Pattern Languages .....	177
7.4	Patterns of Analysis .....	179
7.4.1	The Behaviour Pattern .....	181
7.4.2	The Behaviour Typology Pattern .....	183
7.4.3	The Environmental Behaviour Pattern .....	185
7.4.4	The Organisational Behaviour Pattern .....	186
7.4.5	The Process Pattern .....	189
7.4.6	Patterns of Objects .....	191
7.4.7	Patterns of Organisation .....	198
7.5	The Case for Analysis Patterns - a reprise .....	201
8	Conclusion .....	204
8.1	The pre-eminent cause of system failure .....	204

8.2	The difficulty with requirements .....	205
8.3	The limitations of hard system methods .....	206
8.4	The role of cognitive psychology .....	207
8.5	Towards a sustainable methodological plurality .....	208
8.6	Patterns of analysis .....	209
8.7	The essential features of a corporate knowledge base.....	211
8.8	A new paradigmatic convention .....	212
8.9	Analysis – an unending quest for knowledge.....	213
8.10	Opportunities for further research .....	214
8.11	The Contribution .....	216
9	Bibliography.....	219
10	Original Sources .....	235

## Tables Of Figures

Figure 1-1 A framework for mapping methodologies (Mingers, 1997).....	4
Figure 2-1 Three types of system error (DeGrace & Hulet Stahl, 1993).....	12
Figure 2-2 The 'waterfall' model.....	13
Figure 2-3 Boehm's spiral model (Boehm, 1986) .....	15
Figure 2-4 A dimension-based taxonomy of systems (Checkland, 1981).....	18
Figure 3-1 Approaches to Strategy-making (Hamel & Prahalad, 1994).....	38
Figure 3-2 Simple or single-loop learning (Stacey, 1993) .....	42
Figure 3-3 Complex or double-loop learning (Stacey, 1993) .....	43
Figure 3-4 The 'ladder of inference' (Ross, 1994) .....	46
Figure 3-5 The knowledge creation process (Nonaka and Takeuchi, 1995).....	54
Figure 3-6 The epistemological and ontological dimensions of organisational knowledge creation (Nonaka & Takeuchi, 1995).....	56
Figure 3-7 Four modes of knowledge conversion (Nonaka & Takeuchi, 1995).....	56
Figure 3-8 Five-phase model of the organisational knowledge creation process.....	59
Figure 4-1 The coding and abstraction schema (Boisot, 1995).....	67
Figure 4-2 A semiotic scale of abstraction (Boisot, 1995) .....	68
Figure 4-3 Polanyi's modes of knowing in Boisot's E-space (Boisot, 1995) .....	70
Figure 4-4 Kolb's learning typologies in Boisot's E -space (Boisot, 1995).....	71
Figure 4-5 Boisot's U-space knowledge typology (Boisot, 1995).....	75
Figure 4-6 Boisot's C-space knowledge typology (Boisot, 1995) .....	76
Figure 4-7 Boisot's SLC phases in the I-space (Boisot, 1995).....	79
Figure 4-8 Boisot's SLC in the I-space (Boisot, 1995) .....	80
Figure 4-9 Boisot's entropy in the I-space (Boisot, 1995).....	81
Figure 5-1 The top-down structured design lifecycle.....	97
Figure 5-2 The IE Pyramid (Martin, 1986).....	102
Figure 5-3 The triumvirate of IE architectures.....	103

Figure 5-4 CRC: Process Overview (Coleman et al, 1994).....	112
Figure 5-5 CRC: Class Card (Coleman et al, 1994).....	112
Figure 5-6 Event schema components .....	121
Figure 5-7 The Ptech Event Analysis Method.....	123
Figure 5-8 Object flow diagram (Martin & Odell, 1992).....	125
Figure 5-9 The rational enterprise philosophy (Jacobson et al, 1992).....	128
Figure 5-10 The process transformation (Jacobson et al, 1992).....	129
Figure 5-11 The OOSE system development processes and associated models. (Jacobson et al, 1992)....	129
Figure 5-12 Jacobson's analysis model (Jacobson <i>et al</i> , 1992) .....	131
Figure 5-13 The Models of Object Oriented Development (Booch, 1991) .....	135
Figure 6-1 The End-product as a Capability (from Hale) .....	156
Figure 6-2 A protocol of communication between two processes.....	157
Figure 6-3 An illustration of how the execution of a use case may straddle multiple embedded functional silos. .....	167
Figure 7-1 The three orthogonal dimensions of a business system domain.....	178
Figure 7-2 A process pattern for a trivial use case.....	190
Figure 7-3 A 'diamond-shaped' process pattern for a non-trivial use case.....	191
Figure 7-4 Leavitt's diamond illustrating a socio-technical framework (Leavitt, 1965) .....	200
Figure 8-1 Four classes of organisational behaviour (Batram, 1998).....	204

# **Chapter 1**

## **1 Introduction**

The purpose of business analysis may be considered to be the elicitation and expression of requirements that define a perspective on some business problem for which a solution is required. It is not uncommon for a non-trivial problem to be characterised by complexity and uncertainty, and for comprehension of the problem to be incomplete, inconsistent and inaccurate

The management and IT communities have sought to respond to this challenge by developing 'infallible' prescriptions for analysis. My observations as a reflective practitioner indicate that these prescriptions are far from infallible, and are predicated on simplifying assumptions for which the exigencies of corporate reality provide little or no supporting evidence. While these prescriptions have something to contribute where an environment is stable, they become increasingly untenable as the organisation is drawn deeper into an environment where complexity, turbulence and uncertainty undermine every choice, decision and action.

The motivation for the thesis is to explore the relationship between analysis methods and corporate turbulence. The reasons behind the diminishing effectiveness of specific analysis methods provide the early focus for the thesis. Proposals for equipping analysis methods better to address the challenges presented by corporate turbulence are presented in the later chapters.

The appeal to a broader theoretical base has influenced the structure of the thesis. Many of the chapters approach the analysis paradigm from a particular disciplinary perspective. The consequence is that these chapters contain an element of literature review followed by a definition of how the discipline contributes to a new analysis paradigm. Finally, the contributions are examined collectively to refine the approach taken to analysis by IT and management theorists and practitioners.



Requirements for change to some system are considered generally to provide the focus for analysis. There are few constraints on what is meant by a system. It may be considered to be a market sector, an industry sector, an organisation, a process, a procedure or an IT system. The purpose of analysis may be viewed as defining these requirements with precision and clarity. Chapter 2 presents a challenge to what is understood generally to be a requirement, or at least those interpretations made available to the practitioner community. Evidence is produced that both IT and business initiatives to implement change continue to fail at a rate that indicates there is something fundamentally wrong with our approach. The challenge is that the origin and nature of requirements is the subject of neglect and the victim of simplifying assumptions that are not supported by corporate reality. Chapter 2 concludes by providing a definition of requirements for change that reflects better the dynamics of corporate existence.

The complex nature of requirements enunciated in the previous chapter poses a new challenge to the organisation: how are they to equip their domain experts such that they can apprehend and define these requirements? The theory of organisational learning would appear to provide the answer. The learning organisation has an on-going commitment to developing a repository of information capital that is available to all domain experts. This repository supports the domain expert in identifying and formulating requirements for change. Chapter 3 introduces an examination of various organisational learning models. A recurring theme in these learning models is that knowledge is emergent and any successful learning model must take account of this feature. Reference is also made to the primacy of tacit knowledge and the use of mental models to detect and interpret shifts in organisational behaviour. Knowledge is considered to emerge when it is transformed from a tacit to an explicit state. It is also recognised, however, that the cyclical nature of knowledge creation requires that the expression of explicit knowledge must possess sufficient texture to prompt tacit reflection. Chapter 3 concludes by evaluating the contribution of organisational learning and knowledge creation to our understanding of the origin and nature of requirements.

Chapter 4 provides a more detailed exploration of knowledge creation. Of particular interest here is the stress placed on the abstraction and codification conventions used to construct knowledge assets. Where the conventions are inappropriate, the domain expert is presented with a highly effective barrier that deters any cognitive investment. Similarly, efforts at knowledge sharing among groups of domain experts are thwarted. It is argued, therefore, that an effective analysis method must be based on a system of metaphorical expression built from abstraction and codification conventions appropriate to the domain of enquiry. The implication is that analysis must be based on methodological pluralism. A related issue discussed here is the economic value of knowledge that is considered to be under endless assault as utility and scarcity are eroded. Knowledge, or rather the requirements that emerge from knowledge, therefore have a 'window of opportunity' after which they fail to confer any benefit on the organisation. The conclusion drawn from this chapter is that the analysis of requirements is a precarious endeavour that is highly sensitive to the adopted abstraction and codification conventions. Moreover, requirements must be expressed in such a way that they are better equipped to withstand the joint erosion of utility and scarcity.

An evaluation of the major analysis methods available to practitioners is presented in Chapter 5. The methods are evaluated against the criteria established in the previous chapter. Following the framework for mapping methods developed by Mingers (1997), the scope of methods under scrutiny is broadly aligned with the bottom row of the matrix.

	Appreciation of	Analysis of	Assessment of	Action to
Social world	social practices, power relations	distortions, conflicts, interests	ways of altering existing structures	generate empowerment and enlightenment
Personal world	individual beliefs, meanings, emotions	differing perceptions and personal rationality	alternative conceptualisations and constructions	generate accommodation and consensus
Material world	physical circumstances	underlying causal structure	alternative physical and structural arrangements	select and implement best alternatives

**Figure 1-1 A framework for mapping methodologies (Mingers, 1997)**

The arguments constructed in the previous chapters have sought to establish the proposition that requirements must be drawn from knowledge assets. This proposition, however, pre-dates many analysis methods that were developed for a more stable and less complex world. However, this is not to claim the older, more primitive methods have little, or nothing, to add to the domain of knowledge creation; it is more the case that their contribution is marginal and contingent. Conversely, it is concluded that none of the methods under consideration provides a credible approach to developing a corporate knowledge base from requirements for change. Smith echoes the findings of the previous chapter by claiming that 'without metaphor, thought is dead' (Smith 1983) and it is through the choice of inappropriate metaphors that analysis methods are considered to be singularly deficient. In particular, it is found that the adopted metaphors lack conceptual integrity and utility.

The remainder of the thesis is concerned with offering prescriptions to remedy this deficiency.

The explorations of organisational learning and knowledge creation, described earlier in the thesis, claim that opportunities for knowledge creation occur when experts detect shifts in domain behaviour. Yet almost all analysis methods ignore this cognitive predilection and opt for some other focus for analysis, thus introducing a layer of cognitive indirection and increasing the burden on the domain expert. The

claim presented in Chapter 6 asserts that considerations of structure are completely subordinated to those of behaviour; domain structure exists simply to implement behaviour, i.e. abstraction and codification conventions must be predicated on the behaviour of a domain and not its structure. From this basis, Chapter 6 proceeds by defining organisational behaviour in terms that can be adopted by analysis methods.

It is now claimed that, at this point, the arguments presented in the thesis for behaviour-accented analysis methods based on sound metaphorical conventions establish a paradigm from which the analyst and domain expert can collaborate to create knowledge assets with precision and clarity. The emergence of requirements for change can now be addressed with considerably more confidence. To let matters rest here, however, ignores a key cognitive device available to the domain expert. Domain experts use pattern recognition to inform cognitive strategies, and apprehend and interpret behaviour and structure defining a domain. Chapter 7 examines how pattern theory is increasingly influencing the definition of analysis methods and techniques. The argument is presented that patterns enable analysis metaphors to remain recognisable even when they are subjected to a conceptual shift to accommodate some new threshold of genericity or specificity. This protects the domain expert from the necessity of jettisoning knowledge assets when shifting between cognitive strategies. This is not to deny the possibility of Kuhn's paradigm shattering discoveries or inventions (Kuhn, 1970), but rather to place it in a climate where patterns remove cognitive barriers to penetrating complexity and uncertainty. Finally, it is argued that the use of domain patterns assists the expert to distinguish between those concepts that are essential to a domain and those that are incidental. By identifying similarities (or dissimilarities) between domain concepts, the expert is better equipped to exploit the latent value of existing knowledge assets. Chapter 7 concludes by describing generic patterns that defined totally a pluralistic method of analysis that has been used successfully by me on recent assignments in the banking sector.

Chapter 8 presents the concluding arguments. A concluding contention is that while the purview of analysis may be theoretically convenient, it fails the organisation striving to survive in a predatory climate. The variety of perspectives offered by this interdisciplinary study suggests a fundamental revision to the agenda traditionally assigned to the analyst. To simply confine the analyst to requirements that may be expressed from the pool of explicit knowledge is to neglect those requirements that may actually confer competitive leverage on the organisation. The analyst must focus on expressing requirements that emerge from some synergistic blend of explicit and tacit knowledge. The volatile climate in which a corporation must survive suggests, however, that the analyst's responsibilities do not end with simply expressing current requirements. Making provision for perpetuating the knowledge creation cycle should extend the analysis agenda further. Expressions of current requirements should thus be of sufficient texture to stimulate further abstract reflection and launch a virtuous circle of knowledge creation. Analysis patterns are the device for weaving this texture.

The contribution of the thesis is to challenge fundamentally the nature of analysis. Contrary to the reassuring assumptions of many analysis prescriptions, the domain expert now has to survive in an environment where every choice, decision and action is undermined by complexity, unpredictability and paradox. Analysis can no longer rely on a simple prescription. Analysis must now adopt a methodological pluralism and absorb influences from other disciplines if it is to offer the 'requisite variety' of techniques necessary to tackle complex domains. It is also argued that analysis should no longer be considered a periodic activity, but rather an unending quest for knowledge.

## **Chapter 2**

### **2 The Failure of IT**

#### **2.1 Introduction**

Far from empowering organisations to aspire to greater levels of competitive strength and profitability, IT is portrayed frequently as an agent of corporate stagnation and failure. This chapter is concerned with examining why this has happened and presents a series of issues that may contribute to an appreciation of the reasons behind the failure of IT.

Before embarking on a critique of IT, it is essential to establish that statements denouncing the performance of IT are not merely rhetoric. In fact the evidence of IT failure is abundant and has been well documented by numerous industry commentators and researchers. The chapter commences with a review of the literature and evaluates the conclusions addressing the causes of failure. A core theme of the thesis is that IT fails regularly to deliver viable solutions because, in addition to the more familiar causes, both the business and IT misunderstand the nature of business requirements. It is claimed that conventional approaches to capturing requirements make many simplifying assumptions that are not borne out in practice. Moreover, the identification and definition of requirements may be impeded by the very source of those requirements. The hypothesis that requirements cascade in some way from statements of strategic intent is subject to fundamental challenge; indeed, the very concept of strategic planning is examined critically.

Many approaches to developing IT applications assume that requirements emerge in an orderly way in some deterministic universe. Furthermore, it is assumed frequently that members of the business community, with a little coaxing from the IT community, are capable of articulating requirements that are clear, precise and consistent. This is demonstrably not the case in a corporate world that is showing a propensity towards growing discontinuity and turbulence. IT is now increasingly obliged to deliver

solutions to a corporate environment where requirements are unknown or even unknowable. The chapter concludes by appraising the implications of this phenomenon for the IT community.

## **2.2 The Evidence of Failure**

Landauer provides a wealth of authoritative and persuasive evidence that computers have failed to contribute to improved productivity (Landauer, 1995). A survey of labour productivity growth measured as GNP per hour worked in the United States for the period 1870-1993 shows a stagnation during 1950-1973 and a sharp decline for the period 1973-1993. These periods mark the two phases of introduction of computers to American industry. With commendable neutrality, Landauer concludes that the recent stagnation of productivity does not negate the possibility of computers delivering a positive net effect on work efficiency. However, when comparing the productivity of the United States against other industrialised countries, Landauer concludes that 'early, prolonged and heavy investment in computers and information technology.... has failed to prevent a virtual collapse of productivity growth in the United States or to maintain its relative productivity advantage over other industrialised countries'.

Roach coined the phrase 'computer productivity paradox' from analysis of output per information worker versus output per production worker over a period of time (Roach, 1985). Roach found that even in production industries where better productivity growth had been achieved, output per information worker hour did not improve during the period of increased computer use; this was explained in part by the shift from production workers to information workers. Further studies by Roach revealed that during the 1980s, service sector productivity rose by a mere 0.7% although this sector accounted for more than 80% of the investment in IT. Landauer reports that the situation was worse than indicated by Roach's findings, as Roach had only included hardware costs and neglected the downstream costs of software development, operations and support. However, even using Roach's somewhat misleading figures, it is clear that for the

period 1978-1992 while IT investment increased rapidly, productivity growth actually slowed (Roach, 1992).

Landauer turns to Paul Strassmann for an analysis of the relationship between IT investment and business success. Strassmann studied several categories of organisation to analyse the relationship between business success and IT investment measured as the return on assets (net income over the total worth of the company) as a function of the proportion of gross income spent on IT (Strassmann, 1990). Strassmann was unable to detect any conclusive evidence of a positive correlation between business success and IT investment for the period 1977-1987. A negative correlation always appeared to emerge no matter how Strassmann chose to define business success: return on assets, return on shareholder equity, earnings per share, earnings-per-share growth.

From the studies conducted by Roach and Strassmann, Landauer is left to ponder how much of the post-1960s slowdown in productivity growth could be due to the failure of investment in IT (Franke, 1987; Loveman, 1986, 1990) and concludes that the low return appears to be the missing link in the productivity slowdown puzzle. In probing the productivity paradox a little further, Landauer surmises that IT has fueled expansion but not productivity, i.e. IT has enabled organisations 'to do more work but not work more productively'. For Landauer, computers have been pivotal in expanding the modern economy and yet have acted as agents of productivity stagnation. However, Landauer also notes that there is enormous potential for computer-aided gains in productivity. The major advantages in science and technology are perceived by Landauer to be the primary cause of the downturn in productivity. With such a surfeit of technological capability, the emphasis has shifted from 'seeking a solution to a problem' to 'seeking a problem for a solution'. To adopt the latter emphasis invites a dissonance between problem and solution, and imposes a solution on the user that is usually burdened with redundant capacity that obscures the functionality required by the user and is thus detrimental to productivity.



From this premise, Landauer draws the conclusion that the industry must conduct some appraisal of the way it conducts software design, development and deployment. Landauer notes that 'software-producing organisations have too little knowledge of what a system should accomplish'. At the core of this issue is, of course, the enunciation of the requirements defining the problem to be solved. It is recognised that requirements emerge frequently from some indirect and unreliable source, and any utterance is likely to be of limited validity. Landauer describes the software development process as developing a system that 'performs, in a strictly technical sense, many of the functions imagined useful'. There is little doubt that many a disgruntled user would take issue even with this modest achievement. From this humble platform, some form of interface is added to enable the user to operate the system. Landauer implies that the end product would require such a degree of training that, finally, 'the users are redesigned'. Presumably, users are 'redesigned' for each new system they are required to operate. The source of productivity failures of which Landauer complains is now clear: poorly designed systems that take little account of the tasks and skills of their users, and provide only a partial solution to the problem they were designed to address. Far from gaining in productivity, the user has a new phalanx of burdens to overcome, not least to master the cognitive demands of the system. There is little that can be faulted with Landauer's treatise on the failure of IT, and indeed there is much that many industry commentators, practitioners or researchers would recognise immediately. To reverse the decline in productivity gains resulting from inappropriate IT solutions, Landauer advocates the following approach based on user-centred design:

- The requirements for the system should be analysed by observing and talking to prospective users to determine the objectives to be accomplished and how a system may contribute.
- By working with end-users, conduct an iterative evaluation of design concepts by emulating user tasks.
- Finally, when a concept is fully developed, test the resultant system module with the end-users in the environment in which the module will be executed when in production.

To conduct this user-centred design, Landauer exhorts the developer to learn the user's job, consult the users frequently for opinions and suggestions, use subject matter experts, conduct time and motion studies, and consult normal business records to reveal where important gains may be made. Landauer, however, raises a note of caution that intuition and experience have proved a poor guide in predicting the impact of IT.

Landauer proceeds to claim that by controlled experiment, the IT community can neglect the 'deep truths about physical nature' (presumably this is an oblique reference to organisation behaviour) and concentrate on the 'little practical truths about what helps people and organisations do better work faster'. This approach resonates clearly with the objectives of prototyping and rapid application development. There is the risk, however, that while Landauer's method offers the user the opportunity to externalize requirements that would otherwise remain tacit it may neglect requirements emerging from the wider canvas of the corporate landscape. The essential qualities of congruence and completeness (Pressman, 1992; McConnell, 1996; Kitson and Masters, 1993) may be placed under threat.

### **2.3 Other perspectives on IT failure**

An IT system may be considered to fail when it is unable to offer a required level of support to a business process. Failure may occur immediately a system becomes operational or may be the result of declining utility as the system 'ages' and is marginalised by evolving business priorities. A claim of system failure is based either on an inability to deliver the required functionality, or the neglect of some technical or operational constraint that determines the quality of the system deliverables. DeGrace and Hulet Stahl identify three types of system error that may ultimately result in a loss of functionality:

Goal displacement	Goal displacement is simply where goal B replaces goal A on the false assumption that goal B is equivalent to goal A.
Functional distortion	Functional distortion occurs when a particular user perspective is allowed to inappropriately dominate a set of competing perspectives.
Functional distraction	Functional distraction occurs when a particular feature of the system is allowed to become inappropriately influential.

**Figure 2-1 Three types of system error (DeGrace & Hulet Stahl, 1993)**

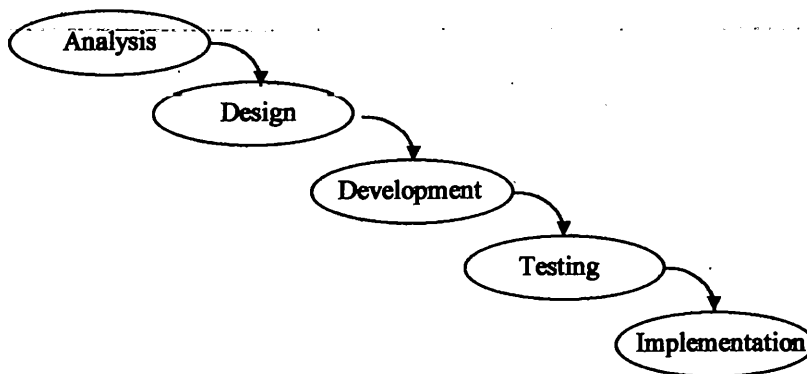
DeGrace and Hulet Stahl argue that to neglect the quality of a system solution, is to invite failure by not giving due consideration to the following issues:

- accuracy
- availability
- reliability
- efficiency
- integrity
- usability
- flexibility
- testability
- portability
- interoperability
- reusability.

The list is a representative, but not exhaustive, account of what are more commonly described as non-functional requirements. Gilb has produced a definitive account of how these system qualities may be quantified and used as a basis for planning and inspection (Gilb, 1988). Indeed, Gilb argues that all requirements can be quantified, so the contribution of a requirement may be measured and some objectivity brought to declarations of system failure.

The sheer size and complexity of many IT system solutions are other obvious causes of failure. There is an understandable tendency to address these issues by acquiring large teams of software engineers to develop solutions. However, in his classic text, Brooks argues against this course of action by pointing out that division of labour and complex webs of communication present their own formidable management problems (Brooks, 1975). For Brooks, the critical issue is to establish and maintain the conceptual integrity of the system deliverable. Brooks warns, nonetheless, that there is 'no silver bullet', i.e. 'no single development, in either technology or in management technique, that by itself promises even one order-of-magnitude in productivity, in reliability, in simplicity' (Brooks, 1987). Brooks would seem to be arguing for a plurality of system development methods selected to deliver a conceptually sound system deliverable.

Many such development methods are based on the 'waterfall' model. The popularity of the paradigm (it has endured for over three decades) is due to its conceptual simplicity and consistency with many trends that have dominated management science until recently.

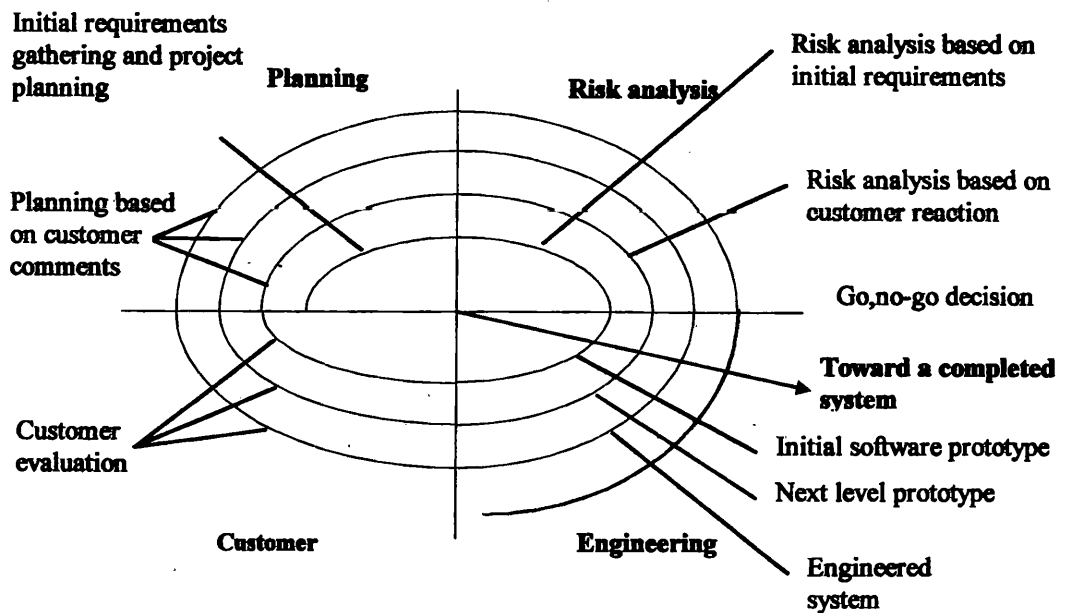


**Figure 2-2 The 'waterfall' model**

Yourdon cites many reasons why the 'waterfall' model fails to deliver the conceptual integrity of the product required by Brooks to avoid the system errors identified by DeGrace and Hulet Stahl (Yourdon, 1992). The 'waterfall model' is considered to include the following weaknesses:

- The sequential configuration of the 'waterfall' model delays the delivery of any results until the later stages of the project.
- The 'waterfall' model depends on stable, correct requirements. Any errors are likely to cascade to successive phases where they are likely to have an escalating effect. As Yourdon points out, the 'waterfall' model can produce a brilliant solution to the wrong problem.
- The thread between requirements and software modules is weakened and may be broken altogether.
- Error detection is reserved for the testing phase. The problem here is that if errors are detected, much analysis, design and development work may need to be repeated.

These weaknesses all derive from the same implicit assumption; i.e. all system requirements are known at the outset of a project. In fact, the converse is true. With the 'waterfall' model, analysis is conducted at the outset of a project when least is known about a domain. No provision is made for requirements that emerge, during the project, with increased knowledge of the domain. Gilb argues persuasively for an evolutionary development of systems based on a spiral iteration of system deliveries (Boehm, 1986). Requirements are allowed to emerge with each iteration of the spiral.



**Figure 2-3 Boehm's spiral model (Boehm, 1986)**

Support for Boehm's spiral paradigm may be found in Lewin's earlier work on social planning where negative feedback defines a circular trajectory in which goals and associated courses of activity emerged during the organisational change process (Lewin, 1947). Lewin considered the organisational change process to comprise the following stages:

- An organisation is considered first to be in a state of stable equilibrium where the forces for change and those opposing change form a 'balanced force field'.
- An environmental event generates a shift that unbalances the equilibrium and forces an 'unfreezing' of the dominant organisational paradigms.
- A period of turbulence and uncertainty persists during the 'reformulation' of a new organisational paradigm.
- Finally, there is a 'refreezing' of the new organisation paradigm as harmony and stability are restored to the new organisational culture now addressing the shift in environmental conditions.

While the spiral model may overcome many of the weaknesses of the waterfall model, it makes no explicit provision for the varying nature of system requirements, i.e. new requirements address specifically the residual risk from the previous iteration. The following section discusses why it is necessary to investigate the nature of requirements a little more deeply.

## **2.4 The Elicitation and Expression of Requirements**

Before embarking on an evaluation of requirements engineering, it may be helpful to clarify precisely what constitutes a requirement. Sommerville and Sawyer define a requirement to be a description of 'how a system should behave, or of a system property or attribute' (Sommerville and Sawyer, 1997). However, they demur from the constraint that a requirement should be a statement of *what* a system does rather than *how* it should do it; the rationale is that it is too simplistic in practice. From this premise they approach the relatively new discipline of requirements engineering for which the following definition is provided: 'all of the activities involved in discovering, documenting and maintaining a set of requirements for a computer-based system'. The term 'engineering' is taken to convey the fact that the process is constituted of systematic and repeatable techniques. They then proceed to the familiar terrain of distinguishing between functional and non-functional requirements only to allege that, in some circumstances, the distinction is unclear. The requirements engineering process is defined as including the following three activities:

- **Requirements elicitation:** the discovery of system requirements through stakeholder consultation and scrutiny of system documents and market studies.
- **Requirements analysis and negotiation:** some formal process involving the analyst and various stakeholders by which requirements are accepted.

- **Requirements validation:** an evaluation of requirements to establish consistency and correctness.

These requirements engineering activities are to be conducted in an environment under the supervision of some change management discipline.

Emphasis is concentrated on the necessity of placing requirements engineering in a process improvement programme. To factor improvement into the process, Sommerville and Sawyer draw on the familiar US Department of Defense's Software Engineering Institute's 'Capability Maturity Model' (Humphrey, 1989). The Sommerville and Sawyer version of the model is a three-level schema:

- **Level 1** - The initial level reflects an undisciplined process with individuals responsible for selection of techniques and tools. Consequently, the delivery of requirements specifications is frequently late and over budget.
- **Level 2** - To achieve the repeatable level, the organisation must define standards and procedures for requirements management, and the production of requirement documents and descriptions. At this level, the delivery of requirements specifications is thus more likely to be on time and within budget.
- **Level 3** - The definable level requires that the organisation has defined a requirements engineering process model based on good practices and techniques. Within this framework, it is possible to evaluate new methods and techniques.

From this foundation of 'practical process improvement', Sommerville and Sawyer are able to enunciate ten top guidelines for successful requirements engineering:

- **Define a standard document structure.**



- Make the document easy to change.
- Uniquely identify each requirement.
- Define policies for requirements management.
- Define standard templates for requirements description.
- Use language simply, consistently and concisely.
- Organise formal requirements inspections.
- Define validation checklists.
- Use checklists for requirements analysis.
- Plan for conflicts and conflict resolution.

From these guidelines and supporting infrastructure, Sommerville and Sawyer define an analysis paradigm that is instantly recognisable to many in the IT community. Regrettably, while the paradigm appears perfectly plausible, it will also generate disquiet amongst IT practitioners.

A definition of a requirement is contingent on what is understood by a system, and yet no definition is provided for a system. Checkland presents the following taxonomy (Checkland, 1981):

Bipolar dimension	Property
Rate of Change	Structural (static) Functional (dynamic)
Purpose	Purposive Non-purposive
Connectivity	Mechanistic Organic

**Figure 2-4 A dimension-based taxonomy of systems (Checkland, 1981)**

The essential properties of a business system may be derived from Checkland's taxonomy. A business system is dynamic, purposive, and either mechanistic or organic. To better understand the nature of a

requirement, the key property of a system is its purpose. Although purpose may have been effectively perverted or concealed through years of system transmutation, it is an untenable notion that a business system is without purpose. The purpose may be defined as the desired goal or intention of the system providing the motivation for engaging the system in some way. Holdsworth argues that purpose is signalled by behaviour and investment, and reflects intention (Holdsworth, 1994). Moreover, Holdsworth reasons that the 'process for succeeding in your purpose is not fixed', and it might be added that purpose itself is also subject to change.

Returning to the definition of a requirement provided by Sommerville and Sawyer, it is thus clearly insufficient to omit any reference to the purpose of the system yielding the requirement. A requirement must contain the specification of a quantitative or qualitative contribution to achieving the purpose of the system. By imposing this constraint, it is possible to confirm the veracity of a requirement and expose other requirements that would otherwise remain concealed.

Critically, Sommerville and Sawyer relax the restriction that a requirement simply specifies what should be done rather than how it should be done and further obscure the issue by permitting functional and non-functional requirements to become co-mingled in some unspecified way under certain circumstances. These concessions appear to be in response to the ever-present demands of expediency and pragmatism. However, for good reason, there is a substantial body of literature (particularly where modelling techniques are used to capture and analyse requirements) that insists on the rigid imposition of these constraints. The underlying philosophy is that the problem and solution domains should remain discrete; to allow some aspect of the solution to contaminate the problem domain serves merely to obscure the problem definition and compound the obstacles to arriving at a successful solution. While a requirement may be expressed in terms of a problem and its desired solution, there is an ineluctable imperative for populating these threads separately and associating them through some form of mapping.

The adaptation of Humphrey's Capability Maturity Model to somehow embed requirements engineering into a process improvement program appears to be a sound contribution from Sommerville and Sawyer. Requirements engineering must clearly be conducted in an environment where there is a canon of approaches and techniques available to address a universe of requirements varying widely in both complexity and stability. Unfortunately, from this promising foundation, Sommerville and Sawyer proceed to define a set of approaches and techniques that are based on heuristic principles to elicit requirements, analyse and negotiate requirements, and validate requirements. Throughout, the emphasis is on the use of natural language to describe requirements, with the use of some form of modelling viewed as a supplementary technique. While this approach may be appropriate for managing simple or stable requirements, it is highly debatable whether it remains effective when requirements become complex, ambiguous or unstable. Some doubt must therefore be cast over the ability of this process to even achieve Humphrey's fundamental criterion of repeatability.

This section has concentrated on Sommerville and Sawyer's publication to highlight the fact that the elicitation and expression of requirements are non-trivial activities. It has been argued that a requirement must verifiably contribute to the purpose of the system from which it originates. Moreover, those elements of a requirement that populate the problem and solution domains of the system must be considered separately; it is imperative that the solution does not corrupt or obfuscate the problem definition. Finally, requirements may be simple or complex, stable or unstable, certain or uncertain, consistent or ambiguous, contingent or absolute, low or high priority. One approach or technique is clearly incapable of optimally addressing wide variations in all these dimensions; thus great care must be taken in the selection and application of the appropriate approaches and techniques.

The actual source of requirements is another issue that may present further challenges to commonly accepted conceptions; this is the subject of the next section.

## 2.5 The Source of Requirements

It has been claimed that requirements cascade in some manner from an edict issued from a strategic planning process. In fact, there is a vast body of literature in the fields of strategic management and organisational behaviour that unequivocally endorses this view. Conventional approaches to management theory include the following approaches to organisational effectiveness:

- Peters and Waterman first proposed the pursuit of excellence as a means of achieving organisational excellence (Peters and Waterman, 1982). The pursuit of excellence is predicated on the assumption that there is 'one best way' to achieve organisational excellence, which provides the motto for what is described more formally as the convergence hypothesis. The organisational environment is surveyed to monitor the consequences of actions and absorb feedback from environmental actors. Mintzberg's concern is that this approach draws an organisation towards what is fashionable rather than what is functional (Mintzberg, 1996). This concern is echoed by Stacey who points out that ideological and cultural values become dominant and pervasive, and thus neglect any environmental signal that is inconsistent with the organisational behaviour (Stacey 1993). The available choices of action in response to the discovery of an environmental signal are necessarily limited. Selected actions evolve through multiple iterations of trial-and-error experiments until the 'vision is realised'. The vision/ideology interpretation of the strategic management process is based on implementing the intentions of the leaders within the organisation.

Many of the organisations identified by Peters as exemplars of the convergence hypothesis have, subsequently, had more than a passing acquaintance with catastrophe. It appears that the formulation of 'one best way' remains an elusive prescription.

- Next there is the congruence approach based on contingency theory that seeks to formulate and select strategic plans according to technically rational criteria. Mintzberg reports that congruence seeks to achieve organisational effectiveness by matching a given set of internal attributes with a variety of situational factors (Mintzberg, 1989). Within this paradigm, the selection criteria for a strategic plan are: the overall **acceptability** and desirability of the consequences to the expectations and aspirations of the major stakeholders of the organisation; the **feasibility** of delivering the strategic objectives; and the **suitability** of the strategy with respect to addressing strengths, opportunities, weaknesses and threats. For each dimension, Stacey identifies the following techniques:
  - **acceptability:**
    - gap analysis (Argenti, 1980)
    - scenario, simulation and corporate modelling (Rowe, Mason, Dickel, & Snyder, 1989)
    - investment appraisal techniques (Johnson & Scholes, 1989)
    - sensitivity and risk analysis (Taylor & Sparkes, 1977)
  - **feasibility**
    - product life cycle (Porter, 1980)
    - the experience curve (Boston Consulting Group, 1972)
    - the product portfolio (Hedley, 1977)
  - **suitability**
    - Stacey records that Porter adapted classical economic theories of market form into a framework for exploring the competitive dynamics between an organisation and market. In particular, Porter developed theories of industry structure and general strategy analysis (Porter, 1980), and value chain analysis (Porter, 1985).

- Contingency theory originates from empirical research demonstrating that success was not correlated to a single set of factors, but rather a multiplicity of factors including the trading environment, the size of the organisation, the available technology, the history of the organisation, and the expectations of the stakeholders (Burns & Stalker, 1961; Woodward, 1965; Lawrence & Lorsch, 1967). In contrast to the 'one best way' ethos of the vision/ideology school of management, contingency theory is based on the ethos of 'it all depends', i.e. an organisation must adapt to cope with the 'contingencies' emerging from shifts in environment, technology, scale, and resources (Child, 1984).

Contingency theory is considered to represent an advance on the convergence hypothesis, but still offers no sustainable prescription for organisational effectiveness. In particular, both hypotheses require a profound appreciation of environmental circumstances. This must be achieved before selection of the appropriate combination of strategy, structure and culture. The implication here is that environmental turbulence will militate against a set of circumstances being sufficiently settled to facilitate the formulation of a stable business strategy. The risk is that a strategy devised from contingency theory will be constituted of 'precapsuled programmes to respond in precise ways to stimuli that never quite occurred as expected' (Quinn, 1980).

- Configuration theory is predicated on the ethos of 'getting it all together'. A configuration is defined to be some constellation of structural, cultural, strategic, control systems and other organisational factors appropriate to a particular environment. Miller offers the following principles (Miller, 1986):
  - There are only a limited number of possible constellations of organisation attributes that are feasible in any environment (Hannan & Freeman, 1977; Aldrich, 1979; McKelvey 1981)
  - Organisations are driven toward a few common configurations in order to achieve internal harmony and consistency between structures, cultures, strategies and contexts.
  - Organisations tend to change the elements of their configuration in a manner that either extends a given configuration or moves quickly to a new configuration that endures for a long time.

The 7S framework (Waterman, Peters & Phillips, 1980) is identified as a typical example of a configuration; the success of an organisation is determined by the configuration of seven attributes collaborating to form some unified system.

The aim of configuration theory is commendable: to transform an organisation into a harmonious, consistent and coherent ensemble. However, configuration theory is undermined by serious weaknesses. Stacey notes that the drive for stability and regularity focuses on a small number of rather standard patterns, and not a large number of separate criteria. Mintzberg reports that configuration theory may be ineffective for both mechanistic organisations and adhocracies. For the mechanistic organisation, the dominant influence of executive declarations may inhibit innovation. Conversely, the 'looseness' of an adhocracy may contrive to prevent a viable innovation from ascending to any position of dominance.

- From Mintzberg's five forms of configuration (Entrepreneurial, Professional, Innovative, Diversified, Machine), it is possible to extrapolate further hypotheses of organisational effectiveness.

Rather than adhere to a particular configuration, an organisation may balance competing forces by combining multiple configurations. As Mintzberg notes, 'Effective organisations usually balance many forces. Configuration merely means a tilt towards one force; combination is more balanced.' The combination hypothesis poses the risk that competing configurations might confront each other immobilising the organisation.

The conversion hypothesis describes the migration from one form of organisational effectiveness to another as progress is made through some lifecycle trajectory. When a conversion is overdue, the migration is not contentious. However, to be overdue is to expose the organisation to competitive threat. Conflict is more likely where opportunity or threat is less clearly discerned.

To navigate a sustainable trajectory, Mintzberg maintains that the organisation must be able to reconcile contradictory forces by achieving a dynamic equilibrium between the cooperative force of ideology and the competitive force of politics. Either force can promote or suppress change.

Disquiet about the ability of conventional approaches to formulate successful business strategies is now widespread. In fact, there is clear evidence that these approaches are failing to deliver sustainable success (Hayes & Abernathy, 1980; Pascale, 1990; Hamel & Prahalad, 1994). The conventional hypotheses of organisational effectiveness all appear to include deficiencies of sufficient seriousness to alert the analyst to the possibility of requirements, emerging from these theoretical foundations, containing significant anomalies. The next section contains a discussion of, possibly, the most potent source of error.

## **2.6 The Strategic Planning Paradox**

Mintzberg offers compelling arguments to explain the failure of strategic planning and even poses a fundamental challenge to the very concept (Mintzberg, 1994). A distinction is made between the various types of strategy:

- intended strategies formulate organisational patterns for the future
- realised strategies identify organisational patterns emerging from the past
- deliberate strategies are those intended strategies that are fully realised
- unrealised strategies are those intended strategies that are not realised.

However, Mintzberg reports that much literature of planning ignores another crucial form of strategy formulation, i.e. emergent strategy: a strategy that was realised but not expressly intended. Alternatively, emergent strategy is yielded from a set of actions that is consolidated over time to become a consistent



pattern of behaviour in response to some environmental signal that was previously neglected. The detection of environmental signals is, of course, essential to the process of planning, or at least formulating accurate forecasts from which to develop plans. A signal is information emitted from an event and may be regular or discontinuous. While forecasting regular events may be facilitated by the analysis of patterns of relationships concerning the event of interest (Makridakis, 1990), the prospects for predicting discontinuous events appear to be less hopeful. Ansoff offered a resolution by speculating that systems could be devised to predict strategic surprises (discontinuities) through the detection of weak signals (Ansoff, 1975). Yet Makridakis dismisses this proposal as an academic idea of little practical value, requiring considerable abilities far beyond present technological value. Clearly, Mintzberg is not persuaded by either of these stances and cites examples of entrepreneurs who have responded to weak signals to achieve considerable success as a means of resolving this apparent impasse.

Mintzberg dismisses the claim of 'commitment planning' (Ackoff, 1970), that strategy may be devised according to a predetermined schedule that totally disregards the dynamics of the organisational environment. By contrast, Mintzberg endorses the view that strategy formation is a dynamic process that should reflect the dynamics of the organisation and its environment (Quinn, 1980; Pascale, 1984; Mintzberg and McHugh, 1985). Thus Mintzberg contends that the assumption of predetermination that informs much literature of planning, is a fallacy.

Mintzberg continues by launching an assault on the assumption that strategic planning should strive to detach strategic management from operations management. It is asserted that effective strategists are those that immerse themselves into daily detail while being able to abstract strategic messages from it. The endorsement of detachment is apparently predicated on a further assumption that environmental fact, noise, inference and impression can be conveyed accurately and completely by hard data. Mintzberg proceeds to dismantle this edifice by arguing that hard data is:

- limited in scope, lacks richness and often fails to encompass important non-economic and non-quantitative factors
- too aggregated for effective use in strategy making
- delivered too late to be of use in strategy making
- simply unreliable.

After enunciating the fallacies of predetermination and detachment informing much literature on strategic planning, Mintzberg's sights are focussed on what is perceived to be the central fallacy: the assumption that strategic planning can be formalised. Mintzberg can find no evidence that formalising strategic planning systems captures the messy, ill-structured processes from which strategic plans are formulated in the real world.

By discrediting the assumptions of predetermination, detachment and formalism, Mintzberg has effectively transformed the edifice of strategic planning into a highly dubious source of requirements. It can be deduced from Mintzberg's observations that requirements emanating from strategic plans are likely to simply address regular signals from the environment. There appears to be no capacity for detecting or articulating responses to weak and discontinuous signals; and yet, apparently, it is these very signals that foreshadow the greatest competitive opportunities and threats. Mintzberg postulates that the planning process has for too long been dominated by analysis which by its very nature has served to thwart the creative process. Analysts are described as people preferring 'convergent deductive thinking, to search for similarities among problems, rather than differences, to decompose rather than to design' (Leavitt, 1975). Mintzberg argues that creativity should be factored into the planning process and proceeds to explore individual intuition as a technique for achieving this. Intuition is contrasted to analysis in that it is described as a higher form of synthesis. Mintzberg further develops the contrast by comparing the reductionist instincts of analysis to consider knowledge as discrete chunks, whereas creative intuition strives to forge knowledge into continuous images. The intuitive universe is one of 'simultaneous, holistic and relational concepts' defined by 'soft, speculative information' that necessarily remains tacit. From this

premise, Mintzberg affirms that strategy making (the predominant source of requirements) is a dynamic, ambiguous, discontinuous, interactive process that emphasises synthesis and learning, and compels the workforce to utilise intuition.

Having sought to challenge commonly held perceptions about the prescriptive and deterministic nature of requirements, the next section addresses their actual origins.

## **2.7 The Nature Of Requirements**

From the previous section it is reasonable to conclude that requirements may emerge from two sources: analysis and intuition. Analysis is conducted in a deterministic, linear universe where the environment emits strong, regular signals that the organisation can forecast with confidence and respond to in some pre-determined way. However, analysis has little to offer where the universe is non-deterministic and non-linear, and the environment emits weak, discontinuous signals where meaning is unclear. The only access to this universe is through intuition and instinct; properties which are notoriously elusive and difficult to articulate. However, this is not to claim that intuition is impenetrable. Forrester argues that intuition evolves through the use of mental models (Forrester, 1975), but that these must be articulated and formalised if they are to be verified for correctness and consistency. Moreover, Forrester highlights significant limitations with intuition: it is poor at assessing the consequences of decisions impacting increasingly complex, multi-loop nonlinear feedback systems. Weick also introduces the possibility of nonlinear relationships causing systems autonomously to change from dominant positive to dominant negative feedback loops (Weick, 1977). To reveal the mental models and thus expose inherent inconsistencies, Mintzberg urges the development of some strategic programming technique such that the strategy, and thus the requirements, may be codified. It is recognised that strategies may be 'rich visions, intricately woven images that may create deep-rooted perspectives' and that the codification should

respect the integrity of strategy to capture every 'subtlety and nuance'. Once codified, the strategy may be elaborated and decomposed progressively into specific action plans.

The origin of requirements is now becoming clearer. Requirements may emerge from analysis of a domain that is well understood by the organisation, or they may be concealed in the intuitive comprehension of a domain for which there is only perfunctory awareness and where elicitation is far from guaranteed. Some codification convention is suggested as a mechanism for exposing requirements and subjecting to them some form of verification.

However, the concept of 'weak' signals suggests the necessity for a more profound and subtle appreciation of the notion of a requirement. The very use of the term 'weak' to describe a signal suggests that the consequences of the environmental event emitting the signal may, in some respect, be disproportionate to the original event. Furthermore, the incidence of 'weak' signals is considered to be discontinuous, i.e. it is difficult, or even impossible, to predict the future emission of 'weak' signals. Finally, the process of extrapolating meaning from the messages accompanying 'weak' signals is often a highly complex and uncertain process, particularly if the messages contradict previous experience of the environment.

## **2.8 Chaos and Complexity**

A somewhat avant-garde movement in management is seeking to follow the example of other disciplines and integrate the principles of chaos and complexity into mainstream management theory. Of particular concern is the inability of mainstream techniques to respond to 'weak' indicators that may signal an episode of upheaval while simultaneously managing the organisation through periods of stability. Stacey is an enthusiastic advocate of this movement and has described in some detail how the theories of chaos (Stacey, 1993) and complexity (Stacey, 1996) should inform management theory.

Awareness has existed for some time that the competitive landscape is characterised by periods of tranquillity punctuated by discontinuous episodes of turbulence and upheaval (Tushman, Newman, Romanelli, 1986). Ormerod detects a similar phenomenon when investigating patterns of unemployment (Ormerod, 1994). It is argued that the periodicity and amplitude of unemployment patterns are determined by the conditions prevailing at the outset of the pattern. Attempts are made to explain comparable behaviour in the financial markets (Cohen, 1997; Mandelbrot, 1998). All such conjecture leads, inevitably, to a discussion of chaos theory. In particular, Stacey seeks to explain patterns of organisational behaviour by drawing heavily on Gleick's account of chaos theory (Gleick, 1988)

Within this paradigm, the stabilising equilibrium of negative feedback loops and the explosively destabilising equilibrium of positive feedback loops constitute the attractors to which an organisation must respond. To be disproportionately influenced by either attractor invites collapse; sustainable survival depends on resolving the tensions between these two attractors. A strategy must be devised to enable navigation of a trajectory between the dynamic borders of these attractors; however, this is to enter an unpredictable universe that is neither stable nor unstable but a paradoxical combination of both (Kets de Vries, 1980; Quinn & Cameron, 1988; Hampden-Turner, 1990).

The application of complexity theory is concerned with an examination of the conditions under which an organisation must devise a survival strategy by adapting to the competing influences of order and disorder. The possible necessity for a duality of purpose may be traced back to the work of Burns and Stalker who noted the distinction between the mechanistic and organic organisations (Burns & Stalker, 1961); the former were bureaucratic, unitary and more suited to stable environments, while the latter were flexible, pluralistic and more appropriate to volatile conditions. The increasing frequency and intensity of turbulence suggests that organisations may no longer opt for a unitary or pluralistic paradigm but, in accordance with Burns and Stalker, must strike a dynamic balance between the competing configurations.

Dominant interpretations of organisational life assume implicitly that success is achieved by restoring an organisation to stable equilibrium once there has been an environmental disturbance (Mintzberg & Waters, 1985; Senge, 1990; Ansoff & McDonnell, 1990). Stacey draws on complexity theory (Prigogine & Stengers, 1984; Waldrop, M 1992; Gell-Mann, 1994) to argue that such an aspiration impedes adaptability. Rather organisations must be thought of as complex adaptive systems collaborating to achieve some purpose (Mueller, 1986; Charan, 1991; Nohria & Eccles, 1992).

Building on this foundation, Stacey defines the legitimate and shadow networks. The legitimate network is based on shared intentionality designed to promote a surprise-free environment exhibiting patterns of behaviour compatible with the organisation's strategic intent. The shadow network is formed from a repertoire of thoughts, perceptions and behaviours that run counter, in some respect, to those generated in the legitimate network (Festinger, Schachter and Back, 1950; Trist & Branforth, 1951; Blauner, 1964; Miller and Rice, 1967). When the legitimate and shadow systems are in conflict, new patterns may emerge to reconstruct the legitimate systems. This hypothesis appears to be endorsed by Mintzberg's description of the creative process that is enacted at the edge and 'far from the logic of conventional organisations.'

An organisation thus behaves as an adaptive feedback network, i.e. a purposive system comprising a large number of interacting agents 'who adjust their behaviour in the light of its consequences for their purpose'. Moreover, an organisation co-evolves with other complex adaptive systems according to some iterative, non-linear trajectory. From this definition, it can be inferred that to make a series of corrective adjustments in response to discontinuous disturbances, the agents comprising the adaptive system must be capable of learning. It is also a property of adaptive non-linear feedback networks that 'specific long-term evolution is radically unpredictable'. Short-term changes are predictable through the judicious use of archetypal patterns of behaviour. However, as Stacey notes, achieving short-term changes through small incremental, progressive strategies is a poor learning strategy, but seeking to resolve tensions between the influences of stability and instability provides a more fertile foundation for organisational learning.

While wholesale adoption of chaos and complexity theory may be considered a little eccentric for the mainstream lobby, to even concede a scintilla of recognition poses a significant challenge to the more conventional theoretical foundations. The influence of chaos undermines the ability of the domain expert to predict the future direction of corporate intent to such an extent that large tracts of the corporate future become unknown and, even, unknowable. The corporation must respond as a complex adaptive system where the corporate future is still difficult to predict, but is occasionally predictable in the short term. The corporation must therefore inhabit a complex rather than a chaotic universe.

To survive in a complex universe, attention should be focussed on using analogy and intuition to detect disturbances in patterns of behaviour rather than seeking specific links between causes and events. This declaration suggests an alternative mandate for requirements engineering.

## **2.9 Holism versus reductionism**

Checkland describes Newton as the methodologist whose principle of reductionism has deeply permeated science for 350 years (Checkland, 1981). There are three senses in which, it is claimed, science is 'reductionist'. The real world is so diverse and complex, that to pursue a particular thread of inquiry, it is necessary to reduce the world by extracting just those features necessary for the investigation. There is an intellectual appeal in achieving logical coherence by reducing an explanation to consideration of an irreducible set of features. Finally, there is the dominant influence of scientific rationalism that argues for scientific thinking to be based on the precept of dismantling phenomena into component elements.

The 'reductionist' paradigm has also acted as a dominant influence in the analysis of business and IT systems. Clearly, to understand fully the architecture of a system, it is necessary to define its characteristics at finer levels of granularity until components are rendered irreducible. Such an argument

has obvious appeal where prescription is essential if deliverables are to be produced according to some schedule.

However, the limitations of the 'reductionist' paradigm have been apparent to the scientific community for some time. A particular source of disquiet is the unsuspected emergence of new phenomena at higher levels of complexity. Further skepticism is introduced by the distinction between 'restricted' and 'unrestricted' science (Pantin, 1968). With a restricted science, far-reaching hypotheses can be evaluated from a limited range of phenomena through 'well-designed reductionist experiments.' Conversely, with 'unrestricted' science, phenomena can be so complex that controlled experimentation is frequently impossible. From the systems perspective, a similar consternation is emerging. There is evidence that as organisations prepare to grapple with environmental turbulence; they are beginning to venture into realms beyond the conventional approaches to analysis. This reasoning poses a challenge to conventional approaches to analysis; they may need to be reconstructed such that they can address unverifiable assertions as well as the more familiar verifiable conjectures.

A system is considered to be holistic when it exhibits a property absent from its constituent elements; Checkland describes this as emergent phenomena. Holism would therefore appear to offer a complementary discipline to reductionism. Rather than simply dissecting a phenomena into a network of irreducible concepts, it is also possible to develop a complementary perspective based on the endless web of interdependencies defining the environment for the phenomena. Holistic investigation is, of course, deeply embedded in the sciences of chaos and complexity that are becoming increasingly influential in determining how organisations are apprehended. Cohen and Stewart warn, however, that holism is perhaps not quite what is needed; there is an apparent tendency to consider a system as a whole and ignore its context (Cohen & Stewart, 1994). The distinction between the two approaches to addressing problems is defined as follows:



- the reductionist pursues a thread of inquiry based on a step-by-step causality seeking to describe a domain in terms already associated with the domain
- the antireductionist is a pattern-seeker striving to describe a domain with terms available from other domains.

Following Cohen and Stewart, the analyst is faced with a dilemma when confronting a non-trivial domain; quite simply is the domain simple or complex. The answer is that it all depends on the context of the inquiry. It is likely that nothing is quite as simple nor as complex as might be believed by first impressions. 'Simple rules can breed simple or complex behaviour; complex rules can breed simple or complex behaviour.'

Therefore, as the analyst is required to confront domains of ever increasing turbulence, complexity and uncertainty, it is clear that the reductionist approach preferred by conventional approaches to analysis is insufficient. The inherent reductionism must be complemented by a holistic quality that empowers the pattern-seeking compulsion in the domain expert.

## 2.10 Conclusion

This section has sought to challenge the assumption that requirements emerge in some orderly manner from the prescribed behaviour and aspirations of an organisation. While this may be appropriate when order prevails, it is argued that to concentrate on this focus may invite dissonance between a corporation and its environment through increasing stagnation. Those requirements arising from the endeavours to negotiate between order and chaos, however, appear to have been neglected by the mainstream literature of requirements engineering. Yet it is this endeavour that is best equipped to notice weak signals in the environment that may presage some escalating disturbance to which the organisation must respond. It could thus be argued that this endeavour recognises the earliest manifestations of competitive opportunity

or threat. However, much of the knowledge from this endeavour exists only at an intuitive or instinctive level where it is necessarily elusive and difficult to articulate. To penetrate this knowledge requires analysis of archetypal patterns of behaviour for which some shift or disturbance in response to a weak signal from the environment may indicate the need for action. To neglect these requirements thus consigns IT to a position where it is endlessly lagging behind the demands of the organisation, which in turn may lag behind the reality of its environment. It is little wonder therefore that IT fails to achieve its objectives.

However, to slavishly respond to the influence of chaos is also to invite organisational dissonance through disintegration. It has been argued that the 'actual' requirements of an organisation emerge from some negotiation of the tensions implicit in the requirements emanating from the competing influences of order and chaos. Moreover, this negotiation is a creative and innovative process that is only possible on the edge of chaos, where there is the potential for self-organisation and the emergence of a new order. Negotiation may be interpreted as a process of harmonising archetypal patterns of behaviour that have been disturbed by some environmental event. This can only be achieved if the organisation learns to accurately interpret (and predict in the short term) messages implicit in environmental events that are subject to endless shift and disturbance.

The arguments presented in this chapter provide a context from which to define requirements for change within the dynamic and uncertain reality of corporate existence and survival.

A requirement is a declaration for change that seeks to adapt the association between an organisation and its corporate environment such that it better aligns with some strategic edict. The declaration may address any combination of the corporate environment, the organisation and any system supporting the organisation. Typically, the purpose of a requirement is to resolve dissonance between an organisation and its corporate environment but less obviously a requirement may seek to create dissonance to exploit an opportunity or deflect a threat.

The ability to adapt is essential to the survival of an organisation where its external environment is characterised by complexity; i.e. the competing forces of order and chaos present the organisation with contradictory and paradoxical signals. Requirements conferring the ability to adapt will only emerge when the domain expert learns to continuously interpret and predict events that signal shifts in the behaviour of the organisation or its environment. The next section examines the dominant trends in organisational learning.

## Chapter 3

### 3 Domain Knowledge and the Learning Organisation

#### 3.1 Introduction

A major contributory factor in the enduring failure of IT is identified, in Chapter 2, as an approach to specifying requirements that fails to reflect the dynamics of the environment from which the requirements originate. In particular, it is argued that an organisation must navigate a trajectory that resolves the tensions between the opposing influences of stability and instability; to drift towards either invites corporate decline through either stagnation or disintegration respectively. Requirements for business change should therefore exhibit some blend of concept and context that avoids either of these destinies.

However, it is also recognised that conventional approaches to systems engineering assume, either implicitly or explicitly, that requirements cascade in some orderly fashion from management edicts that contain some declaration of strategic intent. This assumption presents a fundamental dilemma. The declarations of strategic intent from which requirements emerge rarely take full account of the behavioural dynamics of the organisation and its environment. Strategic declarations are thus likely to contain inappropriate conjectures of linearity and determinism that are unlikely to be sustained continuously. It can be inferred therefore that requirements arising from conventional approaches to strategic management are likely to be diminished in terms of both organisational utility and relevance; i.e. the seeds of IT failure are sown even before the launch of a software engineering project.

Not surprisingly, there is a growing awareness that all is not well with conventional approaches to strategic management. The work of Hamel and Prahalad provides a convincing and detailed critique, and offers a set of principles that define a plausible approach to strategic management that appears to confront

the demons of discontinuity and turbulence (Hamel & Prahalad, 1994). The table below presents the distinctions between their approach and the conventional approaches:

	<b>Conventional Strategic Planning</b>	<b>Crafting Strategic Architecture (Hamel &amp; Prahalad)</b>
<b>Planning Goal</b>	Incremental improvement in market share and position	Rewriting industry rules and creating new competitive space
<b>Planning Process</b>	Formulaic and ritualistic	Exploratory and open-ended
	Existing industry and market structure as the base line	An understanding of discontinuities and competencies as the base line
	Industry structure analysis (segmentation analysis, value chain analysis, cost structure analysis, competitor benchmarking, etc..)	A search for new functionalities or new ways of delivering traditional functionalities
	Tests for fit between resources and plans	Enlarging opportunity horizons Tests for significance and timeliness of new opportunities
	Capital budgeting and allocation of resources among competing projects	Development of plans for competence acquisition and migration
	Individual business as the unit of analysis	Development of opportunity approach plans  The corporation as the unit of analysis
<b>Planning resources</b>	Business unit executives	Many managers
	Few experts	The collective wisdom of the company
	Staff driven	Line and staff driven

**Figure 3-1 Approaches to Strategy-making (Hamel & Prahalad, 1994)**

Hamel and Prahalad's approach to crafting strategic architecture reveals an emphasis on exploiting core competencies and the 'collective wisdom' of the company to identify and evaluate new competitive opportunities within enlarged environmental horizons. Core competencies and 'collective wisdom' are

based on a reservoir of knowledge held by the organisation, and may thus be viewed as the 'intellectual capital' of the organisation. Hamel and Prahalad also stress the importance of competence acquisition and migration, which implies that 'intellectual capital' is in some sense dynamic and should be aligned to the aspirations of the organisation. 'Intellectual capital' may only be infused with dynamism through a process of organisational learning where the knowledge base is endlessly reconstructed with the acquisition of new knowledge and the reinvention of existing knowledge.

Analysis of learning orientations (Nevis *et al*, 1997) confirms that whether an organisation embarks on a strategy based on innovation or adaptation, knowledge provides the foundation for pursuing either strategy. Where an organisation decides to embark on a policy of innovation, the knowledge is developed internally. Conversely, where an organisation follows a policy of adaptation, the external environment provides the source of the knowledge. In practice, of course, an organisation is likely to strive for some dynamic blend of innovation and adaptation that best suits the behaviour and aspirations of the organisation and its environment. To maintain a blend of strategic intention that provides for the organisation to detect and respond to competitive opportunities and threats, there appears to be no alternative to launching an endless programme of organisational learning that refreshes the knowledge base or intellectual capital of the organisation.

This chapter is concerned with presenting an evaluation of various approaches to organisational learning and creating an endlessly evolving knowledge base, and also exploring how these approaches influence the creation of requirements for business change.

### **3.2 Simple and Complex Organisational Learning**

To appreciate descriptions of various forms of organisational learning, it is first necessary to examine the foundations upon which they are based. From the previous chapter, it was argued that organisational

behaviour exhibits many of the features associated with an adaptive feedback network. However, within this mode of behaviour, patterns of behaviour can be unintentional, unexpected or counter-intuitive. Moreover, not all organisational behaviour is goal-seeking or purposeful; there are formative phases when organisational behaviour is concerned with identifying goals and defining purpose. Meaning, purpose and vision emerge from a retrospective interpretation of recent behavioural patterns of the organisation and its environment. In the universe of complex feedback systems, prediction becomes very difficult with the need to maintain loosely coupled systems adding further to complexity and uncertainty. The ability to improvise is valued more highly than the capacity for forecasting in this universe. Clearly, an appreciation of human cognition is essential to gaining some understanding of how domain experts apprehend and interpret the behaviour of organisations as complex feedback systems.

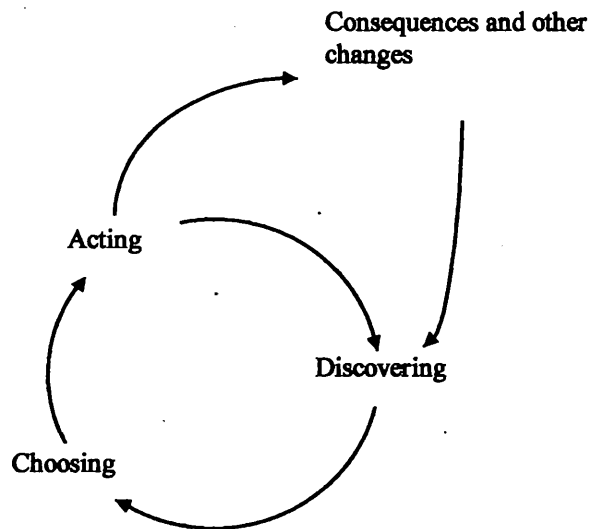
A fundamental assertion of cognitive theory is that human beings create mental models as a mechanism for comprehending and interpreting systems exhibiting excessive size, complexity or uncertainty. A mental model allows an individual to anticipate and make provision for future situations, and utilise knowledge of past events to deal with the present and future. Johnson-Laird advanced the concept of mental models by adding that they are 'analogical representations or a combination of analogical and propositional representations' (Johnson-Laird, 1983, 1988). A mental model is thus some analogical configuration of a set of conceptual objects that reflect the structure of the same objects in the real world. When the domain expert is confronted by some new juxtaposition of behaviour, a reservoir of mental models is used to discern some manifestation of similarity or differentiation, thus enabling more effective deliberation on a new course of action. The domain expert is distinguished from the novice by a superior ability to discern appropriate shifts in the behavioural patterns constituting the mental models. The expert is better equipped at applying analogical reasoning to a more extensive repertoire of behavioural patterns. Analogies are used to solve new problems in terms of old ones (Gick & Holyoak, 1983). As Stacey comments, analogies may be used to either make the novel seem familiar by relating it to prior knowledge or make the familiar seem novel by viewing it from a new perspective.

However, Norman adds a note of caution that mental models are frequently incomplete and unstable (Norman, 1983). Mental models are sketchy, incomplete constructs which are stored and retrieved according to degrees of the similarity and irregularity of vague category features. A domain expert is created when mental models are absorbed by transferring them into the unconscious. However, as mental models exist in the unconscious, they are rarely the subject of dialogue or inquiry, while explicit models are more likely to be questioned. It has already been noted that mental models are formed from analogical constructs by reflecting abstractly on some domain; in addition, mental models are sketchy and incomplete. It is reasonable to infer from these features that any knowledge emerging from mental models is likely to be elusive and difficult to articulate. If the barriers of skilled incompetence (Argyris, 1990) and those arising from the contradictions between the models espoused by experts and those used in practice (Argyris & Schön, 1978) are applied to mental models, it is possible that knowledge embedded in mental models is not only elusive but actually impenetrable.

Mental models are claimed to evolve and approach some threshold of expression as novices and domain experts engage in some behavioural loop of discovery, choice and action within a particular domain. The behavioural loop is considered to correspond to a co-evolutionary feedback process where the actions of one agent impacts a second agent that responds to impact the original agent. From the previous chapter, by definition negative feedback will generate a dampening impact, while positive feedback will generate an amplifying impact.

Simple organisational learning (or single-loop learning) is invoked by negative feedback where any discrepancy between the desired and actual consequences of an action is rectified by some corrective activity (Argyris & Schön, 1978). Single-loop learning is defined as a feedback process from action to consequence to subsequent action that preserves the mental model. The domain expert observes and interprets the consequence of some previous action and chooses a new or adjusted action that is designed to rectify any discrepancy. Negative feedback guarantees that a choice is available that permits a change of behaviour while observing the rules of behaviour.

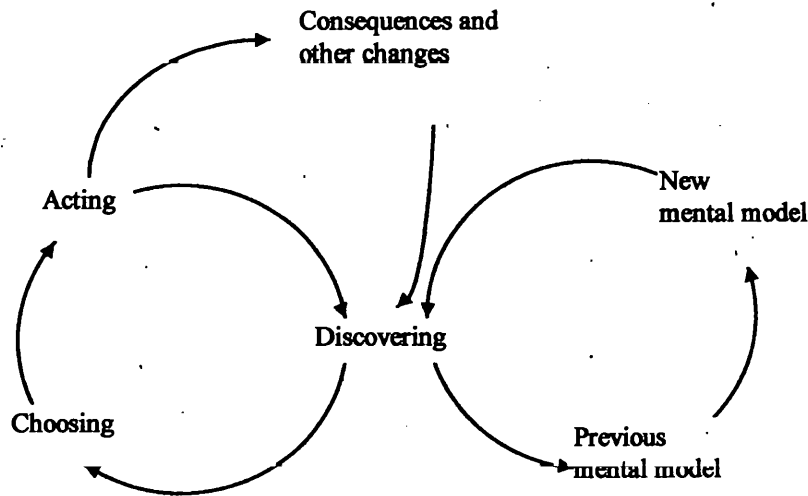




**Figure 3-2 Simple or single-loop learning (Stacey, 1993)**

By contrast, complex (or double-loop) learning is invoked by positive feedback where any discrepancy between the desired and actual consequences of an action provokes a fundamental challenge to the underlying schema (Argyris & Schön, 1978). As with the single-loop model, the domain expert observes and interprets the consequence of some previous action and chooses a new or adjusted action that is designed to rectify any discrepancy. However, positive feedback ensures there is no choice available to rectify the discrepancy and the underlying schema is thus under assault, i.e. the rules of behaviour are changed at the same time as the behaviour itself. Complex learning is thus described as learning in real time or reflection-in-action (Schön, 1987 & 1991). Stacey notes that complex learning involves positive feedback because behavioural consequences must be amplified to destabilise and modify the dominant schema (Stacey, 1996). Unlike simple learning, double-loop learning presents a challenge to mental models and the underlying assumptions that have been responsible for selecting actions. Thus not only actions but also the mental models leading to those actions are subject to change. Complex learning requires the reconstruction of new paradigms from the residue of deconstructed paradigms existing

previously. This process is completely analogous with Kuhn's explanation of how major scientific innovations emerge with the creation of new paradigms (Kuhn, 1962).



**Figure 3-3 Complex or double-loop learning (Stacey, 1993)**

In addition to the previously identified barriers to organisational learning, Argyris reports that organisational defence routines present an additional obstacle (Argyris, 1990). Stacey argues that planned organisational development introduces negative feedback loops to an organisation, while organisational defence routines unintentionally provoke positive feedback loops. Defence routines are considered to include collusion in the avoidance, bypassing, covering-up of contentious issues and games to frustrate the introduction of any innovation that poses a threat to some facet of the established order. Within such an atmosphere, 'dysfunctional learning behaviour blocks the detection of gradually accumulating small changes, the surfacing of different perspectives, the thorough testing of proposals through dialogue' (Stacey, 1993). A commitment model of management is proposed as a mechanism for overcoming organisational defence routines; commitment management is considered to exploit the creativity inherent in domain experts by encouraging public scrutiny of innovations leading to cooperative dynamics and mutual control. However, such an initiative is likely to be resisted as organisations may spontaneously invoke compensatory and defence mechanisms when encountering the forces of change, i.e. positive loops

of behaviour are invoked when levels of uncertainty and ambiguity increase to intolerable levels. Stacey claims that the prospects of success are contingent on the management of the context or boundary conditions of groups within an organisation. The context of a group is defined by the nature and use of power, level of mutual trust, and resource constraints on the agents within the group. The purpose of managing the context is to create an atmosphere of trust and emotional security where defences may be overcome to create an atmosphere that enables complex learning.

From this brief analysis of complex learning, it is clear not only that the knowledge embedded in mental models is both elusive and difficult to express, but also that the circumstances necessary for complex learning are extremely sensitive to organisational defence routines. The atmosphere required for complex learning must therefore be endlessly protected and cultivated by management and those agents active within the organisation.

### **3.3 Triple Loop Learning**

Swieringa and Wierdsma's organisational learning model of triple-loop thinking (Swieringa & Wierdsma, 1992) provides an intriguing extension to the more familiar models of single-loop and double-loop learning. Flood and Romm claim that triple loop learning provides the intelligence and responsibility required to manage the diversity that is becoming an increasingly dominant feature of organisational life (Flood and Romm, 1996). To explain triple-loop learning, it is first necessary to identify the three available centres of learning for single loop learning: Are we doing things right? Are we doing the right thing? Why should we do it?

Flood and Romm comment that the 'Are we doing things right?' centre of learning is concerned primarily with process design and organisational design, thus concentrating on the 'How we should do it?' question. By contrast, the 'Are we doing the right thing?' centre focuses on the 'What should we do?' question and

represents debate based on interpretive thought. The third centre of single loop learning addresses the balance between might and right, i.e. that 'rightness is often buttressed by mightiness, and mightiness by rightness. The concern here is that the mightiness or presumed rightness may obstruct any attempt at organisational learning. Considered in isolation, each centre is seeking a task-oriented resolution and is in conflict with the other centres. As the centres operate in isolation, all cognitive development is non-reflexive and thus of dubious value. Double-loop and triple-loop learning seek to remedy this conflict.

Double-loop learning proceeds by seeking to reconcile the first two centres of learning, i.e. by harmonising 'What we should do?' with 'How we should do it?' and thus challenges the paradigms embedded in mental models. According to Flood and Romm, triple-loop learning is concerned with 'increasing the fullness and deepness of learning about the diversity of issues and dilemmas faced'. This is achieved by establishing a tolerance between the three centres of learning and achieving 'one overall awareness', i.e. a reflexive loop is introduced between the design management and debate management. It is claimed that triple loop learners are reflexive and able to operate more intelligently by looping between the three centres of learning. Flood and Romm comment that might-right management is claimed for triple-loop learning by seeking to establish whether 'rightness is the right of appeal to forms of scientific or presumed agreement, or the might of an appeal to imperatives, that buttress a preferred conception?'

### **3.4 The Emergent Nature of Knowledge**

Single-loop, double-loop and triple-loop learning draw heavily on mental models when deciding how to respond to an environmental event that may be an indicator of change. Where single-loop learning generates a strategy for adaptation without challenging the paradigm embedded in mental models, double-loop and triple-loop learning challenges the paradigm and may result in a revision to mental models used to monitor and interpret organisational events. Recasting mental models provides a new paradigm from which to detect indicators of change, exploit opportunities and deflect threats. However, while knowledge

remains immersed in mental models, it is of limited utility to the organisation. For the knowledge to be of use to the organisation, it must be transformed into an explicit state and thus accessible to a wider audience. The remainder of this chapter examines various approaches to explaining the emergent nature of knowledge in circumstances unencumbered by sociological and organisations barriers.

### 3.4.1 Ross's 'ladder of inference'

From the studies on reflection and inquiry (Argyris & Schön, 1978, Argyris, 1982; Argyris, *et al*, 1985; Argyris, 1990; Isaacs, 1992), Ross applied the metaphor of a 'ladder of inference' to describe the explication of knowledge embedded in mental models (Ross, 1994).

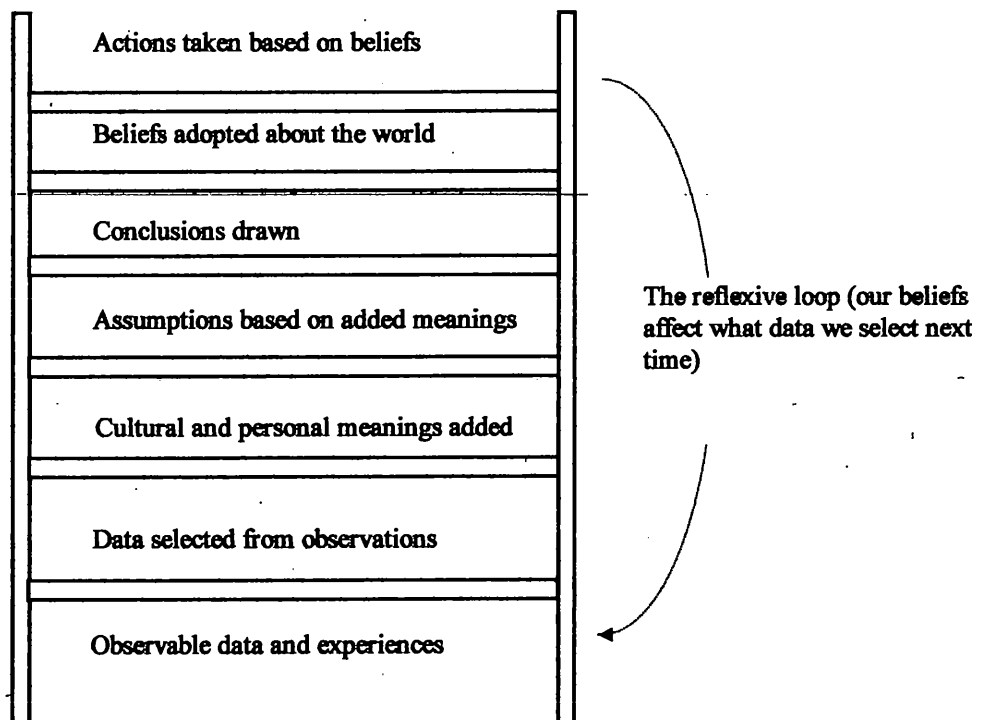


Figure 3-4 The 'ladder of inference' (Ross, 1994)

The 'ladder of inference' metaphor provides a mechanism for describing how knowledge, attitudes and beliefs emerge through a series of inferential leaps from mental models. However, Ross notes that the fabric of mental models and rationale for inferential leaps may have been long forgotten or based on assumptions that were either originally erroneous or no longer relevant to the current domain (Ross, 1994). Three ways in which the 'ladder of inference' can be used to improve communications are identified:

- reflection: increasing awareness of thinking and reasoning
- advocacy: increasing the visibility of thinking and reasoning
- inquiry: challenging the thinking and reasoning of others.

Ross argues that the prospects for evaluating statements emerging from the 'ladder of inference' are improved if the expert is able to identify the data behind the statement and persuade the audience that the data and associated reasoning form a valid basis for formulating consequential abstract assumptions; the key is to align the inference of the expert with the interpretation of the audience.

### **3.4.2 The 'left-hand column' research method**

The 'left-hand column' (Ross & Kleiner, 1994) is based on the two-column research method developed by Argyris and Schön (1974) and provides another method for explicating knowledge embedded in mental models. The purpose of the method is to 'gain awareness of the tacit assumptions which govern conversation and contribute to blocking purpose in real-life situations, and to develop a way of talking about those tacit assumptions more effectively'. This approach to exposing and interpreting tacit assumptions is to select a problem manifesting some degree of intractability and to request every member of the group discussing the problem to record a transcript of their impressions of the discussion. More precisely, the transcript is divided into two columns:

- the right-hand column is used to record the dialogue that actually occurred
- the left-hand column is used to record what the individual was actually thinking in response to each item of dialogue.

The left-hand column of the transcript is examined to discover intention, assumptions, effectiveness and behavioural blocks and thus devise an approach to more effective communication. However, Putman raises some concerns with respect to the effectiveness of the 'left-hand column' method (Putman, 1994b). Both small and large discussion groups generate dynamics that impede the learning process: Putman's response is to recommend the presence of an authoritative and skilled facilitator who is able to promote inquiry and detect the influence of mental models. In fact, the explication of mental models is so sensitive to intentional (and unintentional) obstruction that much attention is paid to developing protocols for balancing inquiry and advocacy (Ross and Roberts, 1994) and defining conversational recipes (Putnam, 1994a).

### **3.4.3 Senge's approach to organisational learning**

The explication of mental models is an essential precursor to the more advanced level of organisational learning that is represented by systems thinking. In addition to endorsing much familiar material on systems thinking, Senge identifies two crucial contributions (Senge, 1994):

- systems thinking requires a qualitative shift from the more familiar linear thinking, i.e. organisations are considered to be constellations of processes and not structures (Capra *et al*, 1991).
- self-organising systems ('where order emerges from chaos') provide a prototype for managing organisations in turbulent environments (Wheatley, 1993).

The adoption by Senge of Wheatley's conclusion (and by implication those of Stacey (1993)) that self-organising systems provide a template for understanding organisational behaviour indicates that this concept is now gaining acceptance in the literature. However, besides including a reference to the concept of self-organising systems, there is little comment on how this may be factored into the theory of organisational learning. Certainly techniques for resolving the tensions arising from stable and unstable attractors are neglected.

Nonetheless, from these foundations, Kemeny *et al* consider effective systems thinking (and thus organisational learning) to be based on simultaneous reflection on: events, patterns of behaviour, systems, and mental models (Kemeny *et al*, 1994). Goodman and Kemeny advocate the use of archetypes as a mechanism for constructing credible and consistent hypotheses for comprehending systems and the associated mental models devised by domain experts (Goodman & Kemeny, 1994). An archetype is defined 'as nothing more than a mental model made visible' and may be used to redesign systems by introducing and removing nodes, and adding loops and breaking links between the nodes. When an organisation is defined by some constellation of archetypes, learning may become focused on breaking through organisational gridlock (Kim, 1993). Through the judicious use of archetypes to identify and define the systemic structures that describe corporate behaviour, organisational learning is achieved by:

- developing a shared vision to inform the redesign of systemic structures
- exploring mental models and team learning to confirm the assumptions underlying organisational behaviour, culture and beliefs
- performing scenario planning to evaluate assumptions about the future
- developing a personal vision and learning to see the world from a creative and interdependent perspective, and not merely from a reactive viewpoint.

Much of Senge's approach to organisational learning is concerned with defining organisational circumstances and attitudes that are conducive to enabling the domain expert to articulate the knowledge,



assumptions and beliefs populating the mental models that reflect some perspective of an organisation. Again, the inescapable conclusion is that to create the sympathetic circumstances necessary for organisational learning, much sustained and reinforced effort is required to overcome entrenched organisational defence routines. It may also be concluded from Senge's study that, for all their flaws, mental models provide a fertile source of knowledge that may be penetrated through the use of archetypes. The concept of knowledge penetration is possibly mistaken. A more likely explanation is that knowledge emerges discontinuously from mental models through a series of inferential leaps. Coupled with this emergence of knowledge is the reconstruction of new mental models from previous unconscious constructs that have suffered some disturbance of their earlier content.

#### **3.4.4 Claxton's hare and tortoise metaphor**

Although Senge's discourse contains much of value, there remains a suspicion that the potential of the unconscious to engage in productive cognitive activity has been underestimated; this is particularly apparent when Senge's observations are compared with those of Claxton (1997). From the perspective of 'cognitive science', Claxton asserts that unconscious intelligence is capable of learning patterns and comprehending problems that are far too complex and subtle for the conscious mind. Claxton's treatise, however, is that the unconscious must be given time to accomplish these tasks and that this proposition runs counter to Western culture which places a premium on conscious, deliberate, purposeful thinking characterised by the following features:

- active thinking is more concerned with finding answers and solutions rather than examining questions
- there is an implicit assumption that there is only one right way of perceiving a problem
- conscious, articulate understanding is the essential basis for understanding, and thought is the essential problem-solving tool
- explanation is valued over observation

- explanations that are 'reasonable' and justifiable are preferred to intuition
- clarity is preferred to confusion which is neither liked nor valued
- a sense of urgency and impatience is predominant
- cognitive activity should be purposeful rather than playful - relaxed cognition is an alien concept
- there is a reliance on precise language that appears to be literal and explicit
- concepts, generalisations, rules and principles are used wherever possible
- there is a maintained sense of thinking as controlled and deliberate rather than spontaneous
- problems that are formed from assemblages of nameable parts can be tackled effectively.

From this last feature, it is clear that Claxton's definition of deliberate thinking conforms to the mode of organisation learning prescribed for responding to stable systems. Claxton recognises that analytical, linguistic approaches may be appropriate for unitary, mechanistic systems, but their utility is quickly exhausted when required to address organic systems. For organic systems, Claxton argues that intuition is an essential complement to reason.

Within Claxton's paradigm, 'knowledge' is defined to be the ability to register patterns and use them to guide future action. 'Learning' is defined as the activity whereby these patterns are detected, and 'intelligence' refers to the resources that facilitate learning and thus the existence of knowledge. From these definitions, Claxton continues by introducing the concept of 'learning by osmosis' to capture the fact that the greater part of the learning process is achieved from acquiring implicit know-how rather than explicit knowledge. Learning by osmosis detects subtle irregularities in experience and uses them to devise an effective course of action. 'The evolution of more sophisticated strategies complements this basic capability; it does not supersede it'. Critically, Claxton concludes that the conscious human intellect stands on the shoulders of learning by osmosis. Apparently, the individual unconsciously detects patterns and formulates appropriate responses without being aware that anything has been learned; although, obviously there has been some degree of implicit learning. Claxton draws on experimental evidence (Lewicki, 1992) to contend that individuals are able unconsciously to detect, learn and use intricate

patterns of information which deliberate conscious scrutiny cannot even see, let alone register and recall. From these studies, Claxton believes that learning by osmosis extracts significant patterns, contingencies and relations from a domain that is so complex that it eludes a conscious, articulate grasp. For Claxton, knowledge emerges from not-knowing. 'Learning - the process of coming to know - emerges from uncertainty'. In accordance with Stacey, Claxton also claims that there is an underlying ambivalence that must be resolved when creating knowledge: learning must seek to reduce uncertainty while also tolerating uncertainty.

By concentrating on the application of deliberate knowledge to a problem situation, Claxton argues that an organisation will generate answers from this singular source 'even when circumstances have changed and new possibilities are there to be found'. This phenomenon echoes simple single-loop learning that introduces the risk of not detecting weak environmental signals that may have an amplifying impact on an organisation unless it is mobilised to respond to the signal. From Claxton's discourse, it is clear that the necessarily fast pace of deliberate thinking only permits single-loop learning, while the more leisurely pace of learning-by-osmosis-enables-double-loop-(and-possibly-triple-loop)-learning. Claxton approaches organisational learning from a different perspective, but has, nonetheless, arrived at a conclusion that is consistent with other studies, i.e. learning can take two forms: deliberate learning (single-loop learning) and learning by osmosis (double-loop learning). Moreover, some balance must be achieved between these modes of learning that reflects the behaviour and dynamics of the organisation and its environment. Claxton's hypothesis is distinguished from other studies by the assertion that unconscious reflection is far more effective than conscious deliberation at interpreting apparently counter-intuitive, ambiguous, contradictory and complex behaviour. Unfortunately, these features now increasingly characterise both organisational behaviour and environmental turbulence.

Consequently, any approach to organisational learning must now address the emergence of knowledge from unconscious cognitive activity based on mental models constructed from metaphors that are both incomplete and inconsistent, and yet may contain patterns of bewildering complexity and sophistication.

Moreover, any disturbance to these patterns may reflect a shift in the behaviour of the organisation that necessitates a decisive competitive manoeuvre. For this reason, it is necessary to consider further the role of unconscious cognitive activity in the emergence of knowledge.

### **3.5 The Primacy of Tacit Knowledge**

From the brief survey of organisational learning presented in this chapter, there can be little dispute that the mental models devised by individuals and small groups of domain experts provide two potent sources of knowledge:

- where a system is sufficiently complex and variable to defy attempts at logical deduction, mental models are used to develop an unconscious understanding of the system
- mental models are used to detect shifts in patterns of behaviour that may presage a competitive opportunity or threat.

It is also clear that knowledge embedded in the deepest recesses of mental models is occasionally capable of ascending through decreasing levels of obscurity until it becomes possible to articulate the knowledge in some form that is meaningful to the organisation. Once this knowledge emerges, it becomes available for integration with knowledge held in the public domain and for exploitation by the organisation.

However, organisational learning is not simply a matter of explicating knowledge held in mental models. There is also the complementary process of disturbances in the public domain generating the material necessary to stimulate unconscious cognitive activity and the consequent evolution of new mental models.

### 3.6 Nonaka and Takeuchi's Knowledge-Creating Company

Nonaka and Takeuchi argue that organisational learning follows a cyclical trajectory (see Figure 3.5) where knowledge may be created in either an explicit or tacit state and may be transformed between those states (Nonaka & Takeuchi, 1995).

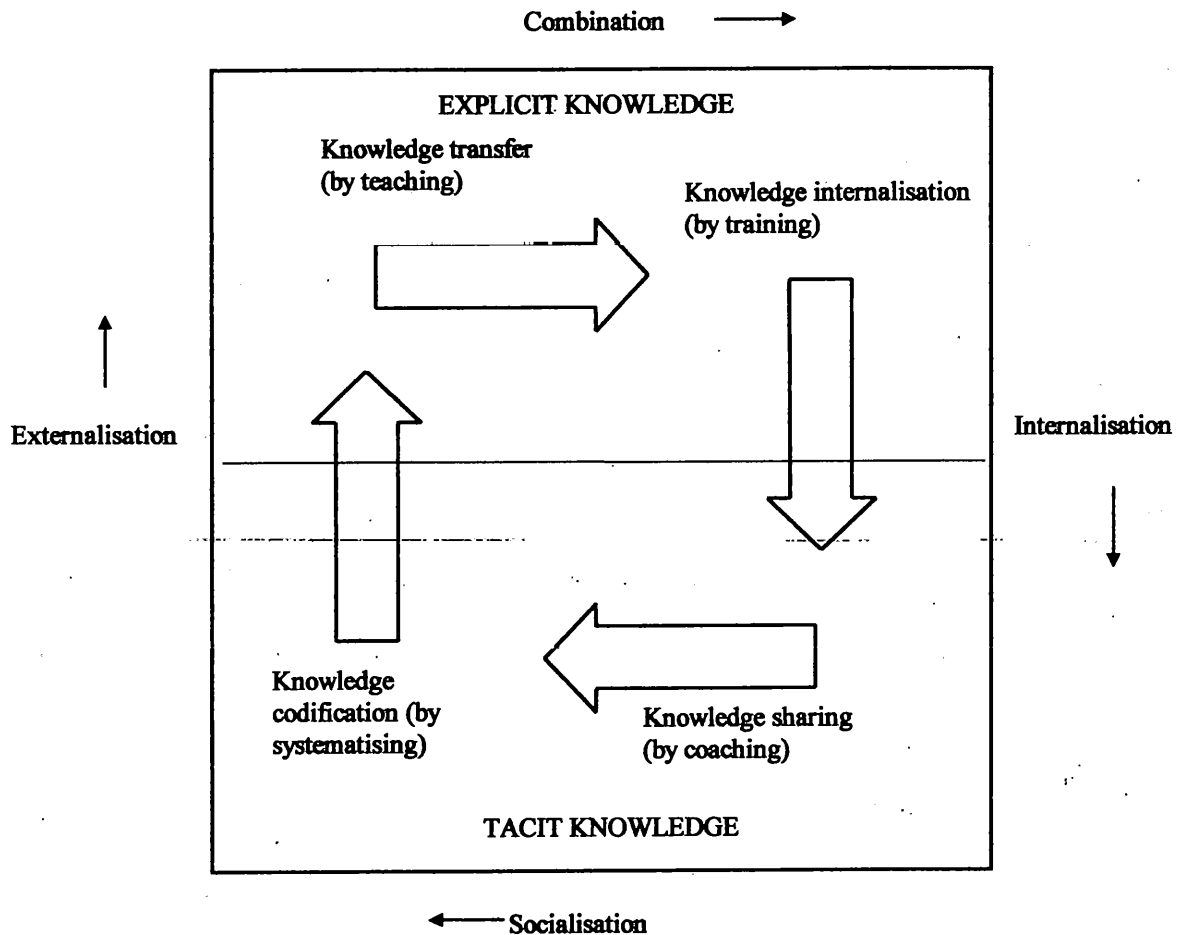


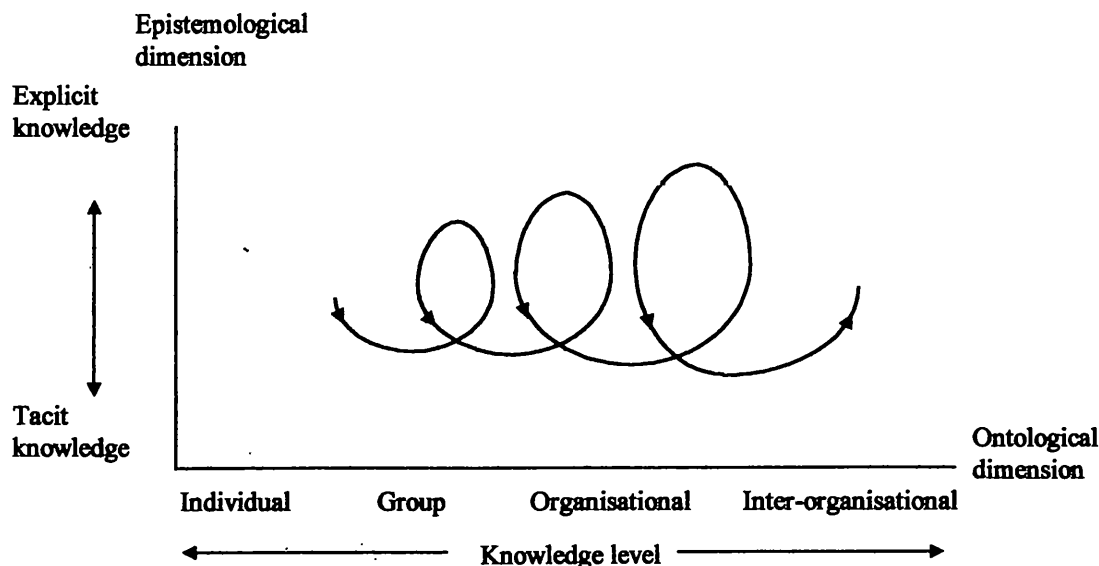
Figure 3-5 The knowledge creation process (Nonaka and Takeuchi, 1995)

Clearly, it is tempting to draw parallels between Nonaka and Takeuchi's use of the terms explicit and tacit knowledge and the earlier discourse on organisational learning. Nonaka and Takeuchi borrow the terms

explicit and tacit knowledge from Polanyi to distinguish between knowledge that is codified and thus transmittable in some systematic language, and knowledge that is personal, context-specific and thus difficult to formalise and communicate (Polanyi, 1966). Polanyi's argument on the importance of tacit knowledge in human cognition is that knowledge is acquired by actively creating and organising individual experiences. Moreover, explicit knowledge represents 'only the tip of the iceberg of the entire body of knowledge'; this sentiment clearly anticipates Claxton's conclusions on unconscious cognitive activity.

Nonaka and Takeuchi are obviously aware of the affinities between their assertions and those of earlier theories of organisational learning, but they make the point that these earlier theories do not incorporate the concept of knowledge creation. Furthermore, Nonaka and Takeuchi take particular exception to the assumptions implicit in Argyris and Schön's theory of double-loop learning. They argue that the theory is predicated on an assumption of 'rightness' and the necessity for intervention from some internal agency; whereas, in reality, organisations continuously create new knowledge by reconstructing existing behaviour, perspectives, culture and beliefs. A broad criticism of literature addressing organisational learning is the absence of any guidance on how to actually create knowledge; by rectifying this deficiency, Nonaka and Takeuchi claim to establish the difference between a knowledge-creating company and a learning organisation.

For the knowledge-creating organisation, knowledge is believed to be created in a cyclical trajectory simultaneously between ontological and epistemological planes, with the spiral trajectory defining the conversion and mobilisation of tacit knowledge. The ontological dimension describes the levels of knowledge diffusion throughout an organisation, while the epistemological dimension defines the distinction between explicit and tacit knowledge.



**Figure 3-6 The epistemological and ontological dimensions of organisational knowledge creation (Nonaka & Takeuchi, 1995)**

The assertion that knowledge is acquired through a spiral interaction between tacit and explicit knowledge enables Nonaka and Takeuchi to identify four modes of knowledge transformation; the possible modes of knowledge conversion are shown in Figure 3.7

	Tacit knowledge	Explicit knowledge
	To	
Tacit Knowledge From Explicit Knowledge	Socialisation	Externalisation
	Internalisation	Combination

**Figure 3-7 Four modes of knowledge conversion (Nonaka & Takeuchi, 1995)**

Socialisation is defined as the transfer of tacit knowledge between individuals through shared experiences and cognitive projection; a process that is referred to more prosaically as 'on-the-job' training.

Externalisation transfers knowledge from a tacit to explicit state and is recognised as the transformation where knowledge creation is manifest and opportunistic innovation may ensue. Norman (1983) makes the point that tacit knowledge is deficient in many crucial respects, and it might added that modes of expression may introduce cognitive barriers. Nonaka and Takeuchi suggest that techniques of deduction and induction are used in an atmosphere of collaborative dialogue and contemplation to elicit and express tacit knowledge in an explicit form. The transmitter and recipient of tacit knowledge therefore have to resort to metaphors, analogies, and paradigmatic constructs to effect a meaningful transfer and conversion. Nisbet (1969), Bateson (1979) and Donnellon, Gray, Bougon (1986) all testify to the potency of metaphors in creating innovative interpretations of experience and concepts. Once a network of metaphors is established, analogies are suggested as a device for harmonising discrepancies by focusing on their structural and functional commonalities. When the metaphors and analogies have stabilised, a conceptual framework emerges from which logical models and prototypes can be developed.

Combination is probably the most familiar form of knowledge acquisition as it associates nodes of explicit knowledge from multiple domains to create new knowledge in an explicit state. By designing new constellations of explicit knowledge, usually through the manipulation of information held on computerised databases, it becomes possible to further exploit the intrinsic value of explicit concepts held in the public domain.

Finally, internalisation is the process by which explicit knowledge is transformed into a tacit state; a conversion that is achieved by an individual considering a node of explicit knowledge in a new context. The conversion requires the individual to embody the explicit knowledge into their prevailing mental models and technical know-how. The act of nurturing this new tacit knowledge until it is sufficiently mature to launch a new socialisation phase creates an asset of considerable value to the organisation.



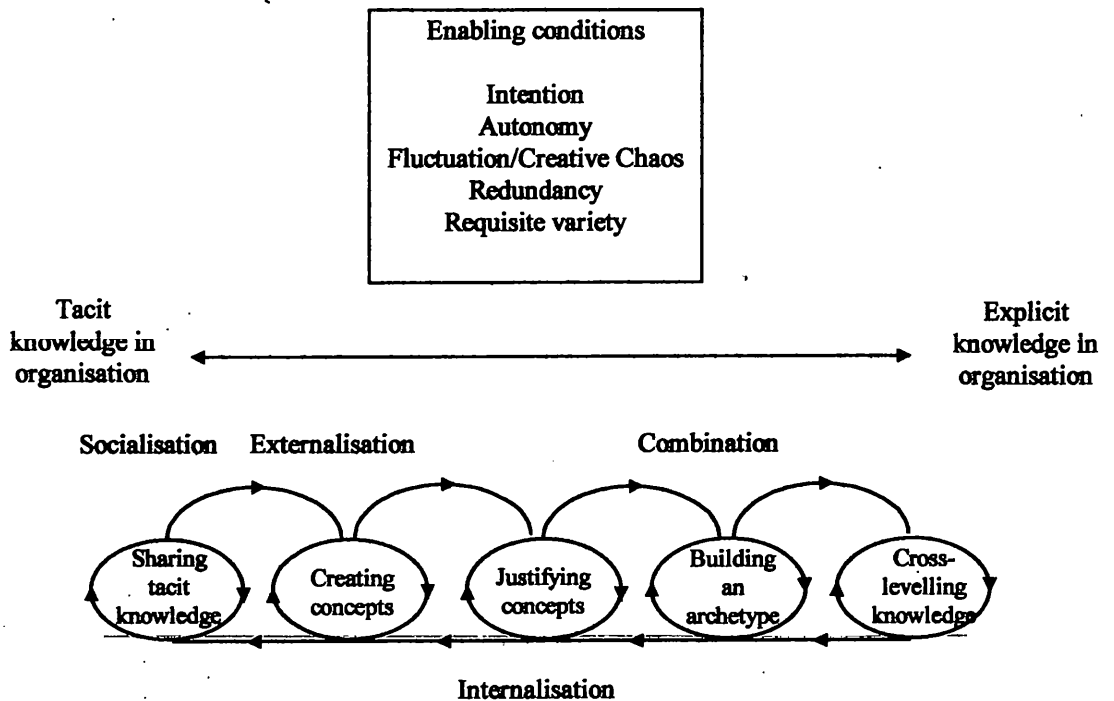
As Nonaka and Takeuchi point out, the socialisation and combination processes simply disseminate knowledge without any change of state and therefore exert only limited influence on the extension of an organisation's knowledge resource; although this is not to discount the importance of diffusion. However, to identify opportunities for the innovative creation of new knowledge, it is necessary to explore the processes of externalisation and internalisation where, hopefully, a vibrant interaction between tacit and explicit knowledge might be detected.

Having described the essentials of the knowledge creation spiral, Nonaka and Takeuchi proceed to identify the enabling conditions for organisational knowledge creation:

- organisational intention (an organisation's aspiration to its goals) establishes the criterion by which an item of knowledge is assessed for validity and utility
- individual autonomy, when practiced judiciously, may improve the prospects of identifying competitive opportunity or environmental threat
- ~~fluctuation and the resultant creative chaos may generate knowledge through interpretation~~ of the interaction between the organisation and its environment as routines, habits and cognitive frameworks are subjected to the ravages of unexpected disturbance
- redundancy of information encourages the overlap of knowledge beyond the existing organisational boundaries and operational needs; such fringe information is frequently a fertile source of knowledge
- finally, according to Ashby, the principle of 'requisite variety' requires that an organisation must possess a level of diversity and variety commensurate with the complexity of its environment (Ashby, 1956); to not achieve requisite variety invites the suppression and displacement of information.

From these enabling conditions, a picture is beginning to emerge of the properties that must pertain to organisational knowledge if it is to provide a legitimate source of requirements for business change. Primarily, the knowledge must be in broad alignment with the strategic intention although this does not necessarily prohibit the reconstruction of inherent perspectives or cognitive frameworks. Secondly,

organisational knowledge may transcend the immediate and obvious needs of the business. Finally, the texture of the knowledge should, in some respect, reflect the complexity and uncertainty of the interrelationship between the organisation and its environment. It is reasonable to expect that requirements for business change emerging from such a context will also exhibit these characteristics.



**Figure 3-8 Five-phase model of the organisational knowledge-creation process**

(Nonaka & Takeuchi, 1995)

Nonaka and Takeuchi submit that tacit knowledge held by individuals forms the basis of organisational knowledge creation and, it might be added, also provides a source of requirements for business change. Accordingly, it is argued that sharing tacit knowledge is the first phase of organisational knowledge creation. However, it has been reported by several sources that tacit knowledge has an unfortunate tendency to defy attempts at simple articulation and is thus difficult to share, particularly among individuals with different backgrounds, perspectives and priorities. To counter these barriers to sharing tacit knowledge, Nonaka and Takeuchi propose the formation of self-regulating teams (there is an obvious

resonance here with the concept of self-organising systems discussed earlier) to engage in some form of dialogue that seeks to synchronise shared experiences and reflections. A note of caution is added in that a self-organising team will only reveal tacit knowledge if the process of socialisation is conducted in circumstances that observe the conditions enabling organisational knowledge creation. Thus socialisation must reflect the requisite variety of the team members and absorb the redundancy of overlapping but variable perspectives. Moreover, the various expressions of tacit knowledge should be verified against all interpretations of organisational intention. It is suggested that management contribute to the efficacy of the group by setting objectives that challenge the received wisdom and, possibly incorporate counter-intuitive propositions. By this technique, sufficient fluctuation may be introduced into accepted cognitive frameworks to generate the creative chaos necessary to stimulate innovation. Finally, management can further empower the socialisation process by conferring sufficient autonomy on team members that they are able to pursue particular lines of inquiry or speculation with some appropriate degree of freedom. The care with which Nonaka and Takeuchi define the circumstances under which organisational knowledge creation is possible endorses the view of many other studies that knowledge creation (or learning) is extraordinarily sensitive to environmental conditions. The clear implication is that any alignment of environmental conditions that does not actively encourage and support organisational knowledge creation will effectively obstruct, and even terminate the process. A neutral environment is thus not sufficient to promote organisational knowledge creation, the environment must actively support the process. Stacey takes up the point when discussing dissipative structures (Stacey, 1993). Such structures (the self-organising knowledge creation team) are unstable and difficult to maintain, and thus require continuous inputs of energy (the continuous endorsement and cooperation of management).

Creating concepts represents the second phase and commences when some consensus has been achieved on the form of the mental models representing the tacit knowledge. The shared tacit knowledge is expressed initially using words and phrases that are progressively transformed into explicit concepts through reasoning techniques such as deduction, induction and abduction. Nonaka and Takeuchi draw attention to the use of abduction that employs metaphors and analogies to express tacit knowledge.

Creating concepts depends heavily on externalisation that is considered to be 'an iterative and spiral process in which contradictions and paradoxes are utilised to synthesise new knowledge'. Much emphasis is again placed on the provision of the enabling conditions for knowledge creation to facilitate the externalisation process.

The next phase endeavours to justify the concepts created during externalisation. Nonaka and Takeuchi note that justification of concepts may be both quantitative and qualitative, i.e. 'the justification criteria need not be strictly objective and factual; they may also be judgmental and value-laden'. However, justification is contingent on the provision of a statement of organisational intent that is sufficiently detailed to enable the emergence of subordinate statements of direction. Moreover, the justification process must acknowledge the fact that environmental turbulence can render the future uncertain and, to some extent, unknowable. Knowledge creation can only counter this phenomenon if organisational intention is also allowed to emerge from the mental models formed by the self-organising knowledge creation groups. The justification process may thus also involve reconstruction of the organisational intention.

Precise construction of archetypes is required to better understand those concepts consolidated during the justification phase. Nonaka and Takeuchi claim that archetypes are constructed by combining newly created explicit knowledge with existing explicit knowledge, and thus building archetypes becomes a process of combination. It is also acknowledged that this process is particularly complex requiring attention to detail and the dynamic cooperation of a variety of departments. The enabling conditions of requisite variety and redundancy of information are identified as key to this process. Finally, it is claimed that a clear statement of organisational intention provides a basis for ensuring that the repertoire of archetypes converges towards a consolidated and consistent schema.

Cross-levelling of knowledge represents the final phase of the knowledge creation process and reflects the act of an archetype ascending to a new cycle of knowledge creation at a different ontological level within

the organisation. Nonaka and Takeuchi contend that this interactive, and spiral cross-levelling of knowledge occurs both within and beyond the organisation. Within the organisation, it is claimed that knowledge that has some physical manifestation or exists as an archetype may create 'a new cycle of knowledge creation, expanding horizontally and vertically throughout the organisation'. Similarly, knowledge created by an organisation may extend beyond the organisation to mobilise knowledge held by affiliated organisations and other stakeholders. Returning to the enabling conditions for knowledge creation, it is argued that an organisational unit must have the autonomy to exploit knowledge accumulated from the organisation and its environment. Arguments are also presented supporting the relevance of internal fluctuation, job rotation, active knowledge transfer, redundancy of information and requisite variety.

From studies into practical knowledge creation, Nonaka and Takeuchi are able to conclude that:

- increases in redundancy and requisite variety are achieved through the provision of up-to-date information
- introducing greater autonomy necessarily involves organisational restructuring
- the infusion of intention and creative chaos is achieved from the effects of fluctuation generated by setting challenging goals.

However, even if it is assumed that management endorses the enabling conditions for knowledge creation actively, there remain substantial hurdles to progressing through the five phases of knowledge creation. The presence of enabling conditions may be sufficient to encourage the sharing of those experiences and perspectives necessary to expose the existence of tacit knowledge. The next four phases, however, are extremely sensitive to decisions governing the mode of expression to articulate the emerging knowledge. The dynamics of knowledge creation ensure that no single mode of expression will capture completely every facet and nuance of emergent knowledge. Equally, it can be conjectured that a mode of expression may exist which captures the essence of the emergent knowledge in sufficient detail that it is able to

progress through the various ontological levels of an organisation and provoke new cycles of knowledge creation. Conversely, it can be inferred that if a mode of expression is immature to the extent that it contains significant incongruity and ambiguity, it may well effectively block the articulation of emergent knowledge or, at the very least, generate a representation of the knowledge that is distorted and otherwise anomalous.

Great care must therefore be taken with the formulation of a mode of expression that is capable both of eliciting emergent knowledge and propagating that knowledge through the progressive ontological levels. The issues surrounding the definition of an effective mode of expression are investigated in the next chapter.

### **3.7 Requirements - a reprise**

Much material has been covered since first addressing the nature of requirements in the previous chapter. It is now an appropriate point to reassess the nature of requirements within the context of the implications arising from the conduct of a learning organisation. The first point to be made is that if an organisation is not committed to a learning culture, it risks decline and ultimate failure from either stagnation or turbulent disintegration. In addition, any requirement for business change emerging from such an organisation is likely to accelerate the decline. An argument central to this thesis is that if an organisation is to prosper (or even survive) it must devise a trajectory that enables it to navigate a route between the fluctuating attractors of stability and instability. To adopt Stacey's metaphor, an organisation must maintain a state of 'bounded instability' that is at the edge of chaos far from stability and instability (Stacey, 1993). In more prosaic terms, an organisation must survey its environment to detect events and interpret signals emitted from those events. If the signals indicate a shift (however subtle) in the essential behaviour of either the organisation or its environment, a judgment must be made as to whether the organisation should adapt to the shift by implementing some appropriate response. Chaos theory is at the

core of this approach because environmental signals may be weak and, if left unattended, can result in escalating consequences that are unexpected, unintended and counter-intuitive. Therein lies the recipe for corporate degeneration.

An organisation may only protect itself from the impact of change by embarking on a continuous programme of organisational learning directed at detecting and responding to environmental signals indicating impending changes in organisational or environmental behaviour. However, Nonaka and Takeuchi take issue with the structural approach of conventional theories of organisational learning and opt for the resource-based approaches of 'core competencies' (Prahalad & Hamel, 1990) and 'capabilities-based competition' (Stalk *et al*, 1992). They argue that organisational learning places too much emphasis on the 'acquisition, accumulation and utilisation of existing knowledge' and tends to neglect the creation of new knowledge. This perspective appears to be entirely consistent with the imperative for an organisation to adapt endlessly to the challenges presented by environmental turbulence; under these circumstances the utility and relevance of existing knowledge must decline with increasing rapidity. As has been argued previously, tacit knowledge provides the conceptual core from which an organisation can elicit and exploit new knowledge. By corollary, tacit knowledge also provides the source of requirements for business change; it is through tacit knowledge that an organisation has the potential to not merely adapt passively to environmental turbulence but to transform itself through interactive participation.

The enquiry into models of organisational learning and knowledge creation reveals that tacit knowledge provides a fertile source of requirements and should be used in conjunction with more expected sources. However, the elusive and impenetrable nature of tacit knowledge and the fragility of the knowledge creation process require that great care be taken with the representation and management of knowledge.

These issues are discussed in more detail in the next chapter.

## **Chapter 4**

### **4 The Knowledge Creation Cycle**

#### **4.1 Introduction**

The previous chapter sought to establish that unless an organisation embarks on a programme of knowledge creation and organisational learning, opportunities for change might be lost. It was argued that many of these opportunities are first manifest as weak signals from either the organisation or its external environment that indicate a shift in behaviour. Moreover, the significance of these signals is often not immediately apparent to the observer. The signal and its meaning are lodged in the unconscious of the observer, and it is only after a period of tacit speculation and reflection that any form of articulation is possible. Unfortunately, there are no guarantees that articulation will take place. Indeed, it appears that the converse is more likely to be true; the fruits of any cognitive activity will remain lodged impenetrably in the unconscious of the observer.

The answer, it would seem, is to place the observer in an environment committed to knowledge creation and organisational learning. Here the observer is encouraged to reflect on the significance of shifts in the behaviour of the organisation or its external environment. This alone, however, is insufficient. There must be at the disposal of the observer a cohesive variety of archetypes, metaphors and analogies with which to express elusive and fragmentary concepts. It is through expertise with the use of archetypes, metaphors and analogies to express the unfamiliar or uncertain, that the observer may be transformed into a domain expert. Also, there is some prospect that domain knowledge may be expressed with some degree of precision and clarity, and thus launch a virtuous circle of knowledge sharing throughout the organisation.

This chapter presents a more detailed exploration of the structure of knowledge and its relationship to the dynamics of the knowledge creation cycle.

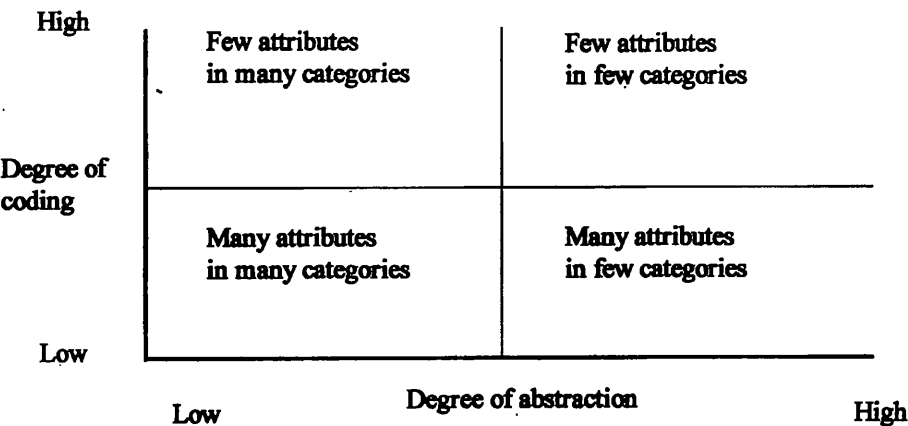


## 4.2 The Structure of Information

In an attempt to explain how knowledge is created and disseminated throughout an organisation, Boisot explores a variety of theoretical perspectives that are based on the structure and communication of information (Boisot, 1995). The exploration commences with the proposition that cognitive activity employs two fundamental techniques to extract information from data: coding and abstraction. Coding is defined as organising an experience of some environmental phenomenon into a perceptual category selected from a repertoire of possibilities exhibiting varying degrees of efficiency with respect to that experience. Abstraction enables the individual to generate concepts allowing the perceptual categories to be managed more efficiently by creating generalisations enabling discrete perceptual and conceptual categories to be manipulated as single entities.

Accordingly, Boisot declares that coding economises on the quantity of data to be processed, while abstraction economises on the number of categories through which data will have to be processed. It is argued that efforts towards greater abstraction share a common motivation with attempts at greater codification: they both constitute an endeavour to economise on the data defining a domain and the effort required to process that data. However, it should also be noted, that Holton argues in favour of uncoded abstract knowledge providing a 'well-spring' of scientific creativity' (Holton, 1986). Moreover, Holton contends that speculation can be guided by themata that are 'universally shared even though scarcely articulated'. Abstraction and codification conventions thus represent a strategy for introducing simplifying assumptions and configurations that better equip the domain expert to confront a deluge of complexity and uncertainty that might otherwise be overwhelming. Successful abstraction and codification conventions must therefore achieve some balance between capturing the essence of a domain while necessarily sacrificing some of its detail. Although abstraction and codification conventions share a common motivation, their individual strategies are different yet complementary. Boisot reasons that codification strategies strive to economise on processing data by reducing the complexity of form, whereas abstraction conventions seek greater economies by reducing the complexity of content. Codification thus proceeds by

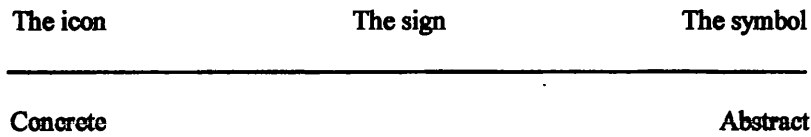
differentiation to enumerate finite sets of discrete elements, while abstraction proceeds by integration to cluster these elements into discrete correlations between individual phenomena.



**Figure 4-1 The coding and abstraction schema (Boisot, 1995)**

Boisot reports that abstract reflection moves knowledge towards greater genericity that may eventually become conceptually adrift from the data giving rise to the original abstractions; under such circumstances, conjecture becomes totally abstract. To penetrate totally abstract concepts, Boisot identifies the necessity for auxiliary theories derived from the dislocated abstractions and yet reflecting to some extent the universe of concrete cognitive assets. If no such auxiliary theories are forthcoming, the knowledge will be consigned to non-disclosure and remain an ‘uncorroborated possibility’.

Boisot claims that coding is the first response when encountering complexity; i.e. coding involves attempting to reduce the number of attributes that have to be considered without losing information by moving up the coding scale towards greater structure. Coding is attempted when a level of noise and ambiguity that is sufficient to undermine existing cognitive frameworks accompanies an environmental signal. The use of abstract concepts greatly reduces the amount of information that requires to be encoded within the cognitive framework. Abstraction may thus be viewed as a choice among competing hypotheses concerning which categories better capture a perceptual attribute.



**Figure 4-2 A semiotic scale of abstraction (Boisot, 1995)**

Boisot argues that coding may occur at different levels of abstraction from concrete experiences with structure, form and content to where conventions regulate the use of symbols and give them theoretical content. It is noted, however, that abstract reflection requires a deeper understanding of the 'causal texture of the phenomena'. Abstract reflection enables a gradual move away from the 'iconic coding' at the concrete end of the abstraction spectrum and towards 'symbolic coding'; at this end of the abstraction spectrum, symbols may be manipulated, stored and retrieved more efficiently because they 'have been drained of perceptual content'. Symbolic coding is considered to be absolutely essential to the efficient processing of information as it reduces simultaneously the quantity of attributes to be managed and the categories to which the attributes may be assigned.

However, the greater efficiencies of information processing offered by symbolic coding are only available to those prepared to make the cognitive investment necessary to acquire the required coding skills.

### **4.3 Polyani's modes of knowing**

Following Polanyi (Polanyi, 1958), Boisot provides the following definitions for three possible modes of knowing:

- The ineffable domain defines a space where knowledge cannot be expressed in any coded form. This space requires no cognitive investment in the acquisition of codes or the mastery of concepts and is

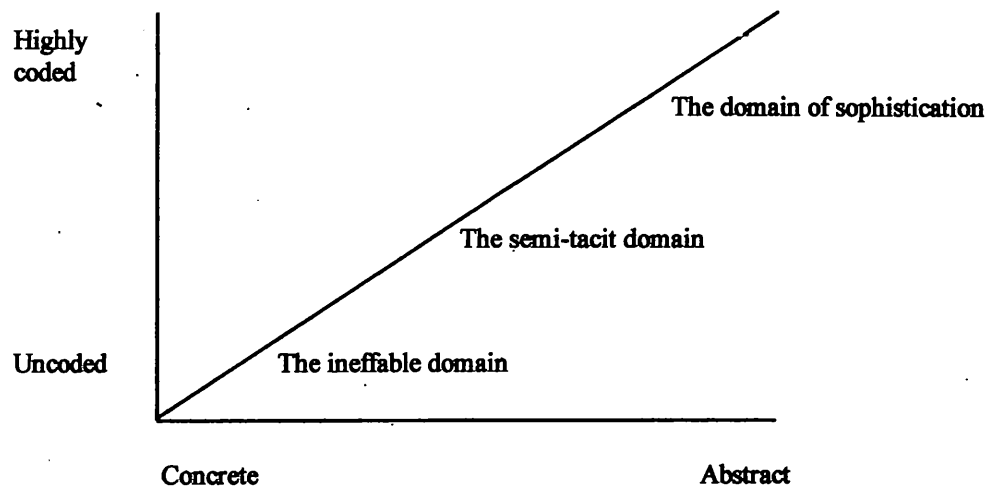
thus, potentially, available to all. However, the absence of any cognitive investment restricts knowledge of the phenomenon to those who were present to experience it.

- The semi-tacit domain provides a space where non-specialised symbols and concepts are used collectively to widely disseminate knowledge. Boisot contends that codes and concepts existing in this space are capable of revealing large expanses of experience to 'comparatively effortless shared understanding', but such sharing is necessarily limited by the cognitive foundations of personal concrete knowledge which is resistant to explication.
- The domain of sophistication draws heavily on highly coded and abstract categories. In this space, tacit and explicit knowledge become disjoint and data embedded in semi-tacit arrays of attributes and categories is 'gradually shed and lost to view'. Polanyi maintains that within this domain, the expert never quite understands what she is articulating since the very act of expression encumbers the tacit element as well as representing it. Novel modes of thought arising from skills at developing and utilising increasingly abstract symbols are considered to have the potential to elude attempts at intuitive understanding.

Boisot maintains that no effort is required to languish in the ineffable domain, but to ascend from this space to the domain of sophistication there must be a cognitive commitment to evolving a framework that contains the necessary codes and concepts. For symbolic processing to chart a trajectory to the domain of sophistication, Boisot draws on the work of Zuboff to note that an individual must overcome the difficulties encountered in operating at higher levels of coding and abstraction, as available cognitive resources may be discarded and new cognitive frameworks created (Zuboff, 1988). It is thus possible to conclude that the information created by symbolic processing may be of considerable value to an organisation but it may only be obtained through substantial and sustained cognitive investment; a conclusion that echoes the findings of Nonaka and Takeuchi.

Boisot conjectures that the symbolic processing required for information structuring is defined by coding and abstraction dimensions that form a two-dimensional epistemological space (or E-space) expressing

the association between abstraction and coding conventions. Learning is defined as moving in the E-space.



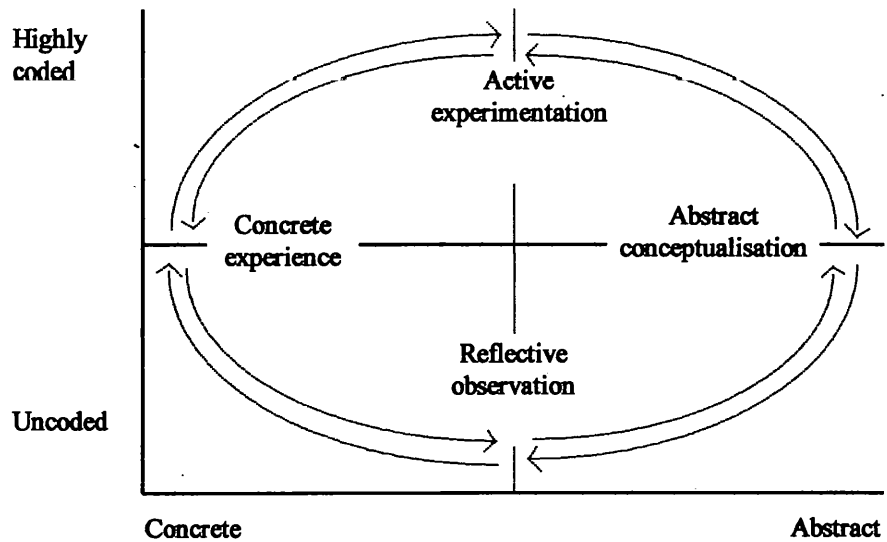
**Figure 4-3 Polanyi's modes of knowing in Boisot's E-space (Boisot, 1995)**

#### **4.4 Kolb's learning typology**

Kolb's experiential learning theory (Kolb, 1976) is based on empirical foundations that are consistent with the structure of human cognition depicted by Boisot's E-space, and is of particular interest because four learning styles are identified which can be superimposed on the E-space.

However, to superimpose Kolb's learning typologies onto Boisot's E-space, it is necessary to reinterpret Kolb's definitions of active experimentation and reflective observation. For Kolb, active experimentation is considered to be externalised physical activity while reflective observation is an analytical activity based on internal representations and symbolic manipulation. Boisot maintains that active experimentation is the conscious manipulation of well-coded 'data complexions' and reflective observation is the detached, non-committal search for patterns at a lower level of coding. Active experimentation is conducted in a universe where concepts may be manipulated without undue risk of dissolution, whereas reflective

observation permits the endless reinvention and reconstruction of concepts to form new cognitive configurations.



**Figure 4-4 Kolb's learning typologies in Boisot's E-space (Boisot, 1995)**

Boisot comments that an individual's cognitive universe is determined, in part, by the E-space trajectories available through the diversity and complexity of the conceptual configurations constructed by the individual. However, growth in an individual's knowledge represents the accumulation of cognitive assets in certain regions of the E-space that make available new trajectories of learning. Yet an individual is most likely to make the necessary cognitive investment in those regions of the E-space where the cost of disruption is least or the prospects of rewards are greatest; in either case, the possibility of extending a cognitive framework presents the risk of discarding previously acquired and cherished constructs.

Boisot further conjectures that not all knowledge in the E-space is 'equally communicable' and the nature of the inhibitors is determined by the location of the knowledge in the E-space. In particular, only cognitive assets that can be contemplated with some degree of coding and abstraction, can be externalised and communicated. It follows that the greater the degree of coding and abstraction available to a cognitive

asset, the easier it will be to identify, express and verify the knowledge. These findings, of course, reflect precisely Nonaka and Takeuchi's observations on tacit and explicit knowledge and the transfer between these states. Knowledge will remain in a tacit state until some form of coding and abstraction becomes available to hint at its existence. As the coding and abstraction conventions mature into some meaningful basis for cognitive investment, the cognitive asset will ascend the various levels of obscurity until it emerges in an explicit state and is available to the organisation.

#### **4.5 Requisite variety for abstraction and codification conventions**

The adopted abstraction and codification conventions must strike a fine balance. In addition to introducing simplifying assumptions that allow the domain expert economy of cognitive investment, the abstraction and codification conventions must also possess the requisite variety necessary to reflect the diversity and complexity of the domain of consideration (Ashby, 1956). Conventions lacking the requisite variety will simply produce incomplete expressions of knowledge, i.e. a barrier to further cognitive activity and investment is presented by immature configurations. Conversely, over-elaborate conventions may also deter cognitive investment by obscuring the underlying cognitive assets. Following Ashby, the abstraction and codification conventions must offer a diversity that matches the variety and complexity of the domain. Requisite variety is essential if the conventions are to be credible and provide some inducement for adoption by the domain expert. Moreover, by achieving requisite variety, it is more likely that prior cognitive investments may be used to encourage the expert to venture into new regions of the E-space.

#### **4.6 The Transfer of Knowledge**

The previous sections have drawn on Boisot's E-space to explore how knowledge creation is dependent on the co-evolution of coding and abstraction conventions. There is also a clear sense in which organisational learning is cyclical, i.e. as cognitive assets mature to some level of explication they are replaced by

embryonic concepts awaiting the next iteration of cognitive investment that in turn awaits the next evolution of coding and abstraction conventions. This section explores how Boisot extends the hypotheses to address the dissemination of knowledge throughout an organisation.

Boisot notes Kolb's observation that an individual in isolation is unlikely to progress through the E-space, i.e. learning is a collective enterprise that is dependent on very different learning styles and cognitive resources. Therefore knowledge creation is crucially dependent on the capacity of an organisation to transfer knowledge. 'Diffusion' is the term used by Boisot to describe the receipt, processing and storage of an item of information by a target population, i.e. diffusion represents the transfer of knowledge within and between organisations. Boisot identifies three issues that influence the effective and efficient diffusion of knowledge:

- How accurately can a given message be transmitted? (The technical problem)
- How precisely does the message convey the desired meaning? (The semantic problem)
- ~~How effectively does the received meaning affect conduct in the desired way?~~ (The effectiveness or pragmatic problem).

The first issue addresses the physical circumstances of the transmission, and is thus independent of the cognitive assets of both the sender and receiver of the message. The semantic and pragmatic content of a message are the standard by which a message is considered to be meaningful; as Boisot notes from Popper, a communication is considered to be meaningful 'if it modifies the expectations that shape behaviour' (Popper, 1983). It is observed that modern communications technology and universal education may contribute to the resolution of technical and semantic problems respectively, but that the pragmatic problems may only be overcome by a mutual sharing of contexts between the sender and receiver of the communication; a condition that represents the major obstacle to the 'effective diffusion of meaningful innovation'.



To further explore the pragmatic problems associated with transferring knowledge in some domain, Boisot presents the following definitions of the extremes of diffusion:

- undiffused knowledge remains with the domain expert either because the knowledge is difficult to articulate or the expert chooses not to disclose it
- diffused knowledge is knowledge that is shared with others.

Clearly, Boisot's definitions have some resonance with Nonaka and Takeuchi's concepts of tacit and explicit knowledge, i.e. tacit knowledge corresponds broadly to undiffused knowledge and explicit knowledge is diffused, to some degree, within the organisation. However, Boisot's theory of knowledge diffusion originates from a perceived inadequacy of a purely sociological explanation of how information flows within an organisational domain. Boisot maintains that exploration of factors such as social organisation and power relationships should be augmented by epistemological considerations. Therefore, Boisot seeks to develop another theoretical thread by considering how abstraction and codification, the dimensions of the E-space, contribute to the diffusion of knowledge. By considering both the sociological and epistemological contexts for knowledge diffusion, Boisot is able to conjecture that 'a blend of technical, semantic, and pragmatic problems ensures that a given item of knowledge at one time and place is not necessarily the same product it might be at another.' To explain how this transmutation might occur, Boisot defines orthogonal relationships between the dimension of diffusion and the dimensions of abstraction and codification respectively. Each orthogonal mapping yields various classifications of knowledge:

- By defining the orthogonal relationship between the abstract-concrete dimension and the diffusion dimension, Boisot is able to introduce the concept of Utility space (or U-space):

Abstract	Esoteric knowledge 2	Scientific knowledge 3
	1 Local knowledge	4 Topical knowledge
Concrete		
	Undiffused	Diffused

**Figure 4-5 Boisot's U-space knowledge typology (Boisot, 1995)**

The space defined by these dimensions is described as a U-space to explore the relationship between abstraction, diffusion and utility. The U-space schema seeks to establish the principle that as an item of knowledge achieves greater abstraction and diffusion, it offers the 'prospects of genuine gains in utility'. However, Boisot also adds a note of caution: if an advance in abstraction is unable to sustain inference that is free from significant anomalies, there will be no gain in utility. Thus adequate diffusion of knowledge is necessary to ensure that it is subject to the appropriate practice, inferential learning, verification and validation.

Knowledge held in Quadrant 1 is both concrete and undiffused, and is thus highly localised and lacking any social utility; it is likely to be both idiosyncratic and parochial in that its creation is likely to draw only from 'epistemological resources held in this region'. Boisot maintains that any increase of social utility can only occur when the experience yielding the local knowledge is contemplated within an extended scope and broader range of potential applications.

When knowledge achieves a migration into Quadrant 2, it gains potential utility and thus the capacity to offer collective benefit. According to Boisot, when knowledge enters this quadrant it achieves economic value through the acquisition of two properties, i.e. utility and scarcity, and therefore becomes subject to appropriation. Furthermore, as the value becomes more manifest, the increase in potential utility provides the motivation for migrating the knowledge to Quadrant 3. Recounting the local knowledge of Quadrant 1 generates the topical knowledge inhabiting Quadrant 4, such that it is elevated to the status of myth and legend, and is diffused through gossip and rumour.

Boisot argues for a continuous movement of knowledge towards Quadrant 3 based on universal abstract knowledge as an essential pre-requisite for a ‘single, widely shared world-view’. However progress towards Quadrant 3 may be impeded where coding weaknesses and ambiguities frustrate abstraction, and discontinuities in the social fabric of the population inhibit diffusion.

- Boisot introduces the Culture-space (or C-space) as a schema for describing the orthogonal relationship between the dimensions of codification and diffusion. According to Boisot, the C-space provides a schema for exploring how different types of knowledge are structured and shared within a domain.

Codified	Proprietary knowledge 2	Public knowledge 3
	1 Personal knowledge	4 Common sense knowledge
Undiffused		Diffused

**Figure 4-6 Boisot’s C-space knowledge typology (Boisot, 1995)**

Quadrant 3 represents public knowledge that Boisot defines to be freely available in the public domain and transacted anonymously. The perceived danger with public knowledge is that it will become woven into the fabric of a society and the resulting inertia will make it difficult to challenge.

The common-sense knowledge held in Quadrant 4 is defined to be a contingent form of public knowledge which is captured through some form of 'idiosyncratic distillation'. Common-sense is considered to be influenced heavily by preferred social value and belief systems, its acquisition is thus likely to be slower and more partial than knowledge drawn from highly coded sources. Boisot observes that while common-sense knowledge is much less codified than public knowledge, it is, nonetheless, equally widespread. The personal knowledge denoted by Quadrant 1 shares the idiosyncratic nature of common-sense knowledge, but has the further disadvantage of lacking a common 'context suitable for discourse'.

Transforming knowledge from a personal to the proprietary form of Quadrant 2 is considered to occur when appeals to the cultural repertoire of codes fail, and the formulation of some idiosyncratic form of personal coding scheme is achieved. Boisot cautions that the proprietary knowledge may be based on poorly formulated perceptual or conceptual hypotheses and may thus be of little value to the organisation. However, if the proprietary knowledge is of some value, it is noted that the scarcity conferred by its undiffused state will place an economic value on that knowledge. Diffusion of proprietary knowledge into the public domain will, however, necessarily erode the scarcity and thus the value of the knowledge.

With the U-space and C-space schemas, Boisot seeks to demonstrate how abstraction and codification influence the dissemination of knowledge within some domain. By integrating the U-space, C-space and E-space schemas to form a new three-dimensional orthogonal space, Boisot strives to consolidate a framework from which to represent the forces that influence human information processing and sharing

activities. Boisot defines the creation of knowledge as acts of codification, abstraction and diffusion applied to information flows in the E-space, U-space and C-space domains.

#### **4.7 Boisot's Information-space (I-space)**

Boisot's conceptual framework, the I-space, provides a configuration for defining the distinctions between data, information and knowledge. The I-space is a unified, three-dimensional schematic comprising orthogonal dimensions for an E-space, U-space and C-space.

By examining how abstraction, codification and diffusion interact to move information through the data field, Boisot contends that it is possible to describe how knowledge is created in some social system, e.g. the knowledge creation organisation. In particular, Boisot maintains that knowledge is created from a clockwise, cyclical trajectory through the I-space that can be decomposed into six distinct components:

- **Scanning** - a leftward movement in the I-space through which diffused data is transformed into idiosyncratic patterns held by individuals and small groups
- **Problem-solving** - an upward movement in the I-space through which these new patterns gain definite form and contour
- **Abstraction** - a movement towards the back of the I-space through which newly codified patterns extend their range of useful applications and gain generality
- **Diffusion** - a rightward movement in the I-space that makes the newly created knowledge available to a larger population
- **Absorption** - a downward movement in the I-space where the newly created knowledge is internalised through repeated use and becomes largely implicit
- **Impacting** - a movement towards the front of the I-space where new knowledge becomes embedded in concrete practices and physical artefacts.

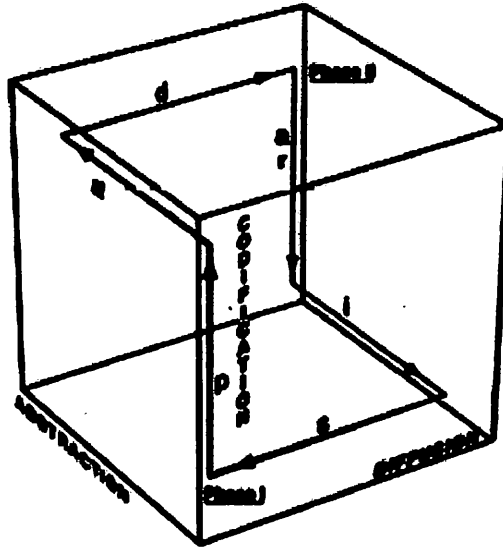


Figure 4-7 Boisot's SLC phases in the I-space (Boisot, 1995)

#### 4.8 Boisot's Social Learning Cycle (SLC)

The phases of knowledge creation described form the SLC. It is noted that the first three phases of the SLC are value generating, while the last three are value exploiting. Moreover, Boisot reports that many SLC trajectories are possible, reflecting the 'opportunities and constraints' imposed by the investment in prior cognitive assets located in the E-space. For Boisot, an effective SLC strategy successfully negotiates 'the constraints and opportunities that it encounters on its journey'.

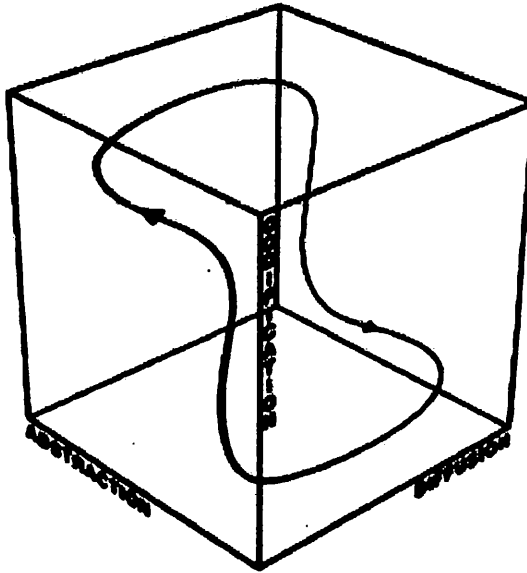


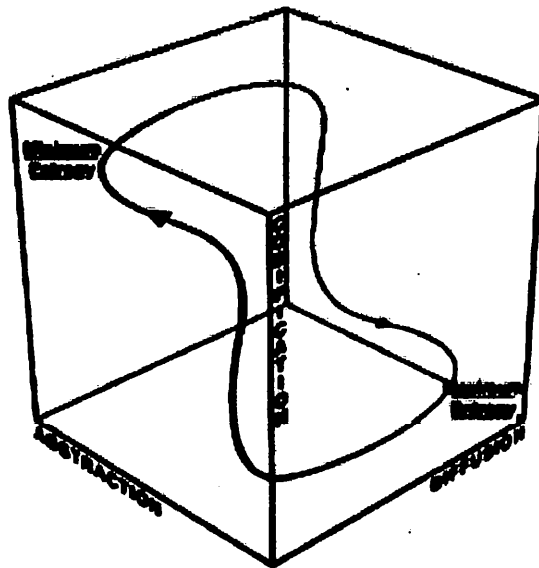
Figure 4-8 Boisot's SLC in the I-space (Boisot, 1995)

#### 4.9 The erosion in the value of knowledge

There is an emerging trend in IT for cherished skills to have an increasingly short 'shelf-life'. The rate of innovation transforms much sought after skills into an asset commanding only marginal interest; the effect of this trend is somewhat ameliorated by the need to support legacy systems. Nonetheless, the message is clear: the value of knowledge is variable. Boisot addresses this issue by employing the I-space to determine when the value of an item of knowledge is at a maximum and at a minimum; for these purposes, Boisot defines the value of an item of knowledge to be a combination of utility and scarcity.

Entropy may be defined as a measure of the disorder of a system, and is the agent through which knowledge loses its value. Following this proposition, Boisot declares that the value of a knowledge asset is declared to be at its maximum when its entropy is at a minimum, and the minimum value occurs when the entropy is at a maximum. Alternatively, value is greatest when uncertainty is at its lowest and scarcity of information is at its greatest; conversely, value is at its least when uncertainty is at its greatest and scarcity is at its lowest.

By using the SLC to explore the fluctuations in the value of knowledge, Boisot develops the following line of reasoning. The instant a knowledge asset achieves maximum value it is subject to the assault of entropy, expressed as forces of diffusion and absorption, endlessly 'eroding whatever value has been created'. Boisot argues that over time, the structured knowledge is progressively exhausted of any inherent scarcity and utility as it shifts towards a position of minimum value. To embark on another cycle of knowledge creation, new knowledge must both be created (from the existing reservoir) and made scarce through evolving conventions of abstraction, codification and diffusion. Boisot's SLC suggests that if knowledge is to be an agent for achieving competitive vigour, the trajectory to minimum entropy must be achieved more quickly than the trajectory to maximum entropy. Unfortunately, within the context of organisational learning, not to embark actively on a process of knowledge creation leaves an organisation unable to both resist a relentless decline to maximum entropy and achieve a subsequent emergence to minimum entropy. However, by attempting to create and exploit organisational knowledge through some learning process, an organisation is better equipped to anticipate periods of maximum entropy and make provision for accelerated ascendancy to another phase of minimum entropy.



**Figure 4-9 Boisot's entropy in the I-space (Boisot, 1995)**



#### **4.10 Barriers to the SLC and the liberating influence of entropy**

In addition to the destabilising influence of entropy, Boisot comments that knowledge creation may be impeded further by blockages preventing progress through the SLC and declares that these impediments may arise from cognitive or social factors, or previous investments in non-synergistic knowledge. Barriers originating from cognitive and social factors can also deflect, distract or otherwise deny information flows, such that traversal of the SLC can be severely constrained or even halted. Boisot raises the intriguing prospect of entropy contributing to the creation of new knowledge assets rather than the more familiar role of marginalising existing knowledge. The argument is that entropy may serve to weaken the edifice of any barrier by endlessly undermining implicit assumptions and beliefs until the dissonance reaches an intolerable peak. The barrier is then deconstructed and from the wreckage it is hoped that the seeds of a cognitive strategy may emerge.

Boisot identifies myriad causes of blockages that may inhibit any of the SLC activities:

- **Scanning** may be blocked by a predilection among domain experts to aligning 'what they see to what they think' rather than adjusting 'what they think to what they see', a condition referred to as cognitive dissonance (Festinger, 1957).
- **Problem-solving** capability may also be influenced by peer-group preferences for selected coding conventions. The dominant coding conventions are discussed in the next chapter which explores the limitations of analysis methods based on various conventions.
- The preference for 'immediacy over reflection' presents a blockage to developing abstracting skills. As Boisot points out, the time required for abstract contemplation is often considerable and the intangible results are greeted frequently with charges of 'lacking pragmatism'.

- **Diffusion** is not only contingent on the semantic richness of the prevailing coding conventions, but also on the ability of the transmitter and receiver to convey and interpret coding configurations unambiguously. In addition, Boisot exposes social barriers that may be introduced by both the transmitter and receiver. While transmitters may seek to exploit some advantage in not releasing knowledge, receivers may resist acceptance of new knowledge based on the perceived probability of threat or disruption.
- A similar sentiment may also present a barrier to **absorption** where new knowledge threatens to disrupt profoundly held beliefs; Boisot argues that this conflict may only be resolved through radical shifts in either the new knowledge or the prevailing system of values and beliefs unless a 'social schizophrenia' is to ensue.
- Finally, Boisot notes that without a physical substrate that may act as a host within which to embed tacit knowledge, **impacting** may be frustrated or even suspended.

#### 4.11 Towards a unified view of knowledge creation

Clearly, the SLC reflects with Nonaka and Takeuchi's cyclical trajectory of organisational knowledge creation. Boisot's conjectures on knowledge diffusion provide another perspective on Nonaka and Takeuchi's description of the progression of knowledge through their ontological dimension. Both approaches declare that the knowledge creation cycle commences with an individual devising some method of expression for knowledge held in a tacit (or implicit) state. While the expression remains tentative and ambiguous at the outset, the articulation matures in some fashion until a domain of relevance can be identified and the knowledge shared with a wider population. Articulation of the knowledge is reliant usually on an extension to extant coding conventions; codes emerging in this fashion enable an expression that is both more precise and less ambiguous, thus facilitating progressive verification and validation of the new knowledge. Once knowledge is formed sufficiently to be subject to this level of scrutiny, it has commenced its diffusion through Nonaka and Takeuchi's various ontological

ontological levels of the population. As the diffusion process develops, the knowledge is subject to progressive maturation and stability, becoming embedded eventually into the processes, practices and culture of the wider population. When the new knowledge is woven fully into the social and technical fabric of the wider population, it is available again for tacit reflection by individuals and small groups, and a new SLC or knowledge-creation spiral commences. This account of knowledge creation is, of course, highly idealised. As Boisot points out, 'many different shapes of SLC are possible in the I-space and most have only a fragmentary and transient existence'; the same comment applies equally to Nonaka and Takeuchi's knowledge-creation configuration.

In common with Nonaka and Takeuchi, Boisot forcibly endorses the necessity for an organisation to embark on knowledge creation rather than organisational learning if it is to achieve sustainable survival. Moreover, both accounts identified similar themes and priorities considered essential to the knowledge creation process:

- ~~knowledge exists in two-dimensional space defined by an epistemological and an ontological dimension~~
- knowledge is created by a movement in this space
- knowledge creation follows a cyclical trajectory through the space defined by these dimensions
- a knowledge creation trajectory may be blocked by the intrusion of cognitive, social and organisational barriers.

In addition to these points of commonality, Boisot further explores the theory of knowledge creation by considering the following issues:

- the epistemological dimension may be extended into another two-dimensional space by introducing an abstraction and a codification dimension

- knowledge may thus be considered to exist in a three-dimensional space defined by abstraction, codification and ontological dimensions
- how the individual creation of knowledge and the subsequent diffusion of knowledge may be blocked by cognitive barriers arising from the inappropriate choice of abstraction and codification conventions
- how the economic value of knowledge is subject to immediate and endless erosion once it achieves an explicit state
- how erosion in the value of a current knowledge asset may provide an opportunity for the creation of new knowledge.

It is argued that the reservoir of organisational knowledge provides the source of requirements for business change. It is also clear from theories of organisational knowledge creation expounded in this section that knowledge must be sustained in a state of endless reinvention. Such a mandate is contingent, in part, on resolving Boisot's extensions to the theory of organisational knowledge creation. Thus Boisot's extensions may provide criteria for evaluating methods of analysis with respect to the elicitation and expression of requirements for business change.

#### **4.12 The Implications for Analysis Methods**

Codification and abstraction conventions are two 'distinct yet interrelated ways of economising on cognitive effort'. Boisot declares that codification provides an expression for a particular form, while abstraction seeks to extend the applicability of a particular form. Through evolving codification and abstraction conventions, it is argued that experience of phenomena becomes more 'organised' and thus achieves a lower entropy quotient. If considerations of cognitive, social and institutional barriers may be put aside, Boisot also argues that knowledge that has achieved a highly codified and abstract state may be more easily disseminated throughout an organisation and acquires a greater economic value. However, at the point of optimal economic value, knowledge comes subject to progressive erosion of value as the

cognitive configurations become absorbed, and thus dissolved, by a wider population. From Boisot's conjectures, it is reasonable to infer that requirements for business change remain elusive until appropriate modes of expression are developed. Moreover, once requirements have found expression, the ever present erosion in their utility and scarcity ensures they have a limited 'window of opportunity' and are thus a perishable commodity.

In common with other approaches embracing the discontinuities of organisational behaviour, Boisot considers the knowledge creation cycle to pursue a trajectory that is both creative and destructive. This apparent paradox is reflected by Stacey's definitions of extraordinary management (Stacey, 1993) and the conflict between 'legitimate' and 'shadow' organisations (Stacey, 1996). In particular, a requirement for business change may originate from an item of knowledge that is epistemologically incompatible with the reservoir of existing knowledge and thus dislodge cognitive investments maintained by the organisation. Such an item of knowledge may therefore erode or weaken the cognitive configurations, culture, beliefs and values that define the ethos of the organisation; it may be anticipated that such an item of knowledge will encounter a measure of cognitive resistance.

From an epistemological perspective, it is argued that requirements emerge from the externalisation phase of knowledge creation, i.e. the phase where knowledge is transformed from a tacit to an explicit state through the processes of scanning, problem-solving and abstraction (the value creation activities). However, it is recognised also that knowledge is likely to remain in a tacit state while extant codification and abstraction conventions lack the maturity necessary to formulate a cogent mode of expression to capture the underlying meaning. To achieve any reasonable prospect of analysing an item of knowledge to elicit requirements for business change, it is thus essential to devise codification and abstraction conventions that the domain expert is prepared to adopt both as an agent of speculation and an effective means of articulation. According to Boisot, this may require the domain expert to abandon cherished cognitive investments; such a decision is only tenable if the domain expert is to be rewarded with the

ability to generate new knowledge with greater efficiency. Such a condition poses a not inconsiderable burden on the conventions presented as analysis tools.

The erosive influence of entropy imposes a fundamental imperative on the organisation. While striving to exploit the economic value of the explicit knowledge, the organisation must also be preparing for the moment where the value is exhausted. The internalisation process, of course, achieves this. Unfortunately for the organisation, the dynamics of the market are imposing increasingly aggressive schedules on the internalisation process. For an organisation not to have to endure a reversal, it must be in a position to exploit a new item of explicit knowledge before the economic value of an earlier item is exhausted. This comment is, of course, purely illustrative; an organisation is formed from a complex web of interdependent knowledge creation spirals, each yielding items of knowledge at different rates of explication with varying quotients of potential economic value which are eroded with a momentum determined by the environmental response.

The competitive vigor of an organisation depends on the efficient execution of the knowledge creation cycle. While this may be inferred directly from Boisot's hypotheses, it should also be noted that, as a corporate aggregate, externalisation must yield new items of knowledge before the internalisation of existing knowledge is fully exhausted. In fact, the period of internalisation defines the 'window of opportunity' during which existing knowledge assets can confer any value on the organisation. Upon termination of the period of internalisation, the knowledge asset ceases to have any relevance to the organisation and is likely to have a detrimental effect if it is acted upon. It is in this sense that requirements for business change are a perishable commodity. Boisot's analysis has thus indicated another, and possibly the most significant, reason for IT failure: in addition to the familiar issues of producing requirements' specifications that are clear, complete, consistent and precise, there is the further consideration of the implications of implementing requirements after their moment is past.

From Boisot, it becomes possible to define a dual agenda for the analysis of requirements for business change: the models, codification and abstraction conventions must support both the externalisation and internalisation processes. To achieve this analysis is presented with the paradox. While a knowledge asset is shifting from minimum to maximum entropy it has the potential to confer economic value on the organisation. Yet during this period while it continues to confer value, it must be challenged to stimulate the internalization process.

#### **4.13 The role of models in managing complexity**

It is generally accepted that any non-trivial domain is likely to represent a system that is beyond the ken of an individual domain expert. Any symbolic coding convention must therefore seek to resolve the fundamental tension between concealing complexity and sacrificing its essential detail. A way forward is to develop abstract models that introduce the simplifying assumptions necessary to achieve a shared understanding of the multiple facets of a domain. 'A good model must behave sufficiently like the real system to allow fairly accurate predictions about the real system's behaviour' (Lovelock, 1991). Following this reasoning, some form of modelling orthodoxy should be devised that integrates the selected abstraction and codification conventions into a cohesive cognitive framework. The modelling orthodoxy must neither become bereft of expressive capability when confronting a complex concept, nor offer such a surfeit of expressive capability that a 'creep' of marginal complexity is permitted to obscure the essence of the domain.

#### **4.14 Towards a new agenda for analysis**

The enquiry into the origin and nature of requirements presented in Chapter 2 concluded that 'requirements that actually confer benefit on an organisation will only emerge when it learns to continuously interpret and predict events that signal shifts in the behaviour of the organisation or its

environment'. The subsequent investigations into organisational learning and knowledge creation have raised a number of issues that challenge the commonly accepted views of what it means to analyse requirements. A contribution of the thesis is to synthesise these issues into a coherent description of the challenges facing the analyst.

- An organisation must seek to adapt to the fluctuating forces of stability and instability; to err on either side is to invite problems.
- A requirement may emerge from a holistic or reductionist scrutiny of a domain.
- A requirement for change is usually in response to a shift in the behaviour of an organisation or its external environment. The response, hopefully, is to adapt in some way that protects future flexibility without sacrificing robustness.
- A behavioural shift can be detected by a 'weak' signal emitted from some organisational or environmental event. 'Weak' signals are difficult to detect and yet can escalate in effect to offer significant opportunities or threats. To ignore 'weak' signals is to neglect a potent and fertile source of requirements for change.
- The front-line workforce is well placed to observe these 'weak' signals. It appears, though, that any interpretation of these signals takes place in the unconscious of the observer where it can remain in an elusive and tacit state.
- The prospects for eliciting tacit knowledge are enhanced if the organisation is committed to organisational learning and knowledge creation.
- Knowledge creation is a cyclical process where knowledge shifts between tacit and explicit states. The process is highly dynamic and precarious, and can experience many detours when navigating between these states. The process is abandoned frequently before either state is achieved.
- Knowledge assets are produced from an iteration of the knowledge creation cycle. Requirements for change emerge from an analysis of knowledge assets.
- An observer is transformed into a domain expert when a cognitive strategy is devised for navigating successfully between these states.



- Knowledge sharing is a key objective of knowledge creation. Unless knowledge is shared effectively throughout groups of domain experts and through the wider organisation, opportunities for knowledge creation will be lost. Modes of expression for knowledge assets should therefore support a cognitive strategy for the group without neglecting the needs of the individual domain expert.
- Archetypes, metaphors, analogies, models and patterns are key elements in a successful cognitive strategy.
- The abstraction and codification conventions adopted to express these elements are critical to the success of a cognitive strategy. Selection of inappropriate conventions may result in the abandonment of a cognitive strategy and the alienation of the domain expert.
- Knowledge begins to lose utility and scarcity immediately it achieves an explicit state. Consequently, there is a limited 'window of opportunity' during which a knowledge asset can confer benefit on an organisation. The organisation should counter this threat by having an endless commitment to refreshing its intellectual capital. The domain expert must therefore be enabled to engage in a continuous process of externalisation (from tacit to explicit knowledge) and internalisation (from explicit to tacit knowledge).

The final three points indicate how conventional approaches to analysis may be impacted by the enquiry into the origin and nature of requirements for change. It is these points that form the basis for redefining the analysis agenda in the remainder of the thesis. The next chapter applies these points to an evaluation of the methods and techniques generally available to the practitioner analyst.

## **Chapter 5**

### **5 The Limitations Of Analysis Methods**

#### **5.1 Introduction**

Analysis methods and techniques form an essential part of the toolkit with which the analyst collaborates with the domain expert to create knowledge assets and thus derive requirements for change. From the previous chapters, it was argued that:

- where a domain is complex or uncertain, knowledge assets are formed from the manipulation of archetypes, metaphors, models and patterns
- great care must be taken with the selection of abstraction and codification conventions adopted to express knowledge assets
- a virtuous circle of externalization and internalisation is required to combat the erosion in the utility and scarcity of a knowledge asset.

It is against these criteria that the efficacy of those methods generally available to the practitioner analyst will be judged. Methods are selected primarily on the basis of their adoption by the user community and therefore some, quite worthy, methods may be omitted.

#### **5.2 A note on systems**

While methodologists strive commendably to introduce clarity and precision, they sometimes introduce ambiguity with the basic concepts from which they construct their methods. One such concept is that of the system for which a range of definitions is available. Darnton & Darnton (1997) provide a series of

definitions of a system. An early definition of a system was provided by Bertalanffy (1968): a system is 'a complex of interacting components, concepts characteristic of an organised whole such as interaction, sum, mechanisation, centralisation, competition, finality, etc., and to apply them to concrete phenomena'. Later, the object property relationship approach to systems allowed Teichroew et al (1980) and Hall (1989) to consider a system as a set of objects (elements or parts), with relations between them and their attributes (properties, or qualities). Moreover, the objects were considered to form an environment with other inter-related objects. Klir simply defined a system as 'a set of things with relations on those things' (Klir, 1991). An alternative view of systems is provided by Checkland (1981) who discerns hierarchies in the formation of systems, which suggests there may be some merit in adopting a holistic course of inquiry when analysing systems. Finally, Beer (1985) followed by Espejo and Harnden (Espejo and Harden, 1989) introduce the concept of systems being recursive at many levels.

Where this concept is undefined, the focus of some methods remains uncertain.

### **5.3 The dominant trends in the analysis of systems**

Before embarking on an evaluation of analysis methods, it is important to clarify exactly what is understood by an analysis method. An issue emerging during the formative years of developing analysis methods was the distinction between the terms 'method' and 'methodology'. For Stamper, 'a method is a specific way of approaching and solving a problem', while 'a methodology is a comparative and critical study of methods' (Stamper, 1988). Although acknowledging Stamper's concerns, Jayaratna reports that within the field of information systems, the terms 'method' and 'methodology' are, for all practical purposes, interchangeable (Jayaratna, 1994). The following definitions of a methodology illustrate the evolution of the concept.

- A methodology will lack the precision of a technique but will be a firmer guide to action than a philosophy. Where a technique tells you 'how' and a philosophy tells you 'what', a methodology will contain elements of both 'what and 'how'. (Checkland, 1981)
- A methodology is a coherent collection of concepts, beliefs, values and principles supported by resources to help problem solving groups to perceive, generate, assess and carry out, in a non-random way, changes to an information situation. (Avison and Wood-Harper, 1990)
- A methodology is an explicit way of structuring one's thinking and actions. Methodologies contain model(s) and reflect particular perspectives of 'reality' based on a set of philosophical paradigms. A methodology should tell you 'what' steps to take and 'how' to perform those steps, but most importantly the reasons 'why' those steps should be taken, in that particular order. (Jayaratna, 1994)

From these definitions, Jayaratna raises the following issues. A methodology exhibits structural properties which may change to reflect interpretation, knowledge, attempts to sequence steps for intervention, and the situation characteristics. The reasons for the ordering of the set of activities or steps as implied by the methodology must be supported by a convincing rationale that achieves the required transformations. The methodology must embrace the concept of a 'system'; Jayaratna notes that a methodology will take either an ontological or epistemological view of a system. Provision must be made for the world image (or *Weltanschauung*, see (Checkland, 1981)) of the user of the methodology. Jayaratna evaluates three threads of methodological development:

- The ETHICS (Effective Technical and Human Implementation of Computer-based Systems) adopts a socio-technical approach to 'creating work systems that are beneficial in human as well technical terms' (Mumford, 1983a, 1983b, 1995, 1996). Mumford defines three basic principles for the ETHICS method

- To enable future users of a new system to play a major role in its design and assume responsibility for designing the work structure that surrounds the technology.
- To ensure that new systems are acceptable to users because they increase both user efficiency and job satisfaction.
- To assist users to become increasingly competent in the management of their own organisational change so that this becomes a shared activity with the technical specialists and reduces the demand for scarce technical resources.

According to Jayaratna, the ETHICS method possesses serious deficiencies with respect to defining the 'problem situation' and managing the social processes for the involvement of the intended problem solvers. From the point of view of the thesis, there are no structured methods to support the investigation of a problem domain, and the prognosis appears to be based on a notional system derived from existing 'taken as given' systems. Unless an analysis method provides for an existing system to be fundamentally challenged, the opportunities for knowledge creation are severely curtailed. For this reason the ETHICS method is not considered any further in the thesis.

- Jayaratna classifies the Soft Systems Methodology (SSM) (Checkland, 1981, Checkland and Scholes, 1990) as an 'issues-oriented' method (Jayaratna, 1988). Jayaratna notes that whereas structured methods are concerned with achieving a single 'truth' state, SSM encourages its users to search for many states, each with potentially the same 'truth' value. In contrast to application of the 'hard' systems engineering methods, i.e. the structured methods, to 'well-structured' situations, the original version of SSM was designed to confront 'ill-structured' situations. According to Jayaratna, this distinction afforded SSM a special status. However, much to the consternation of Jayaratna, later versions of SSM have withdrawn from this position. Jayaratna's evaluation of SSM commences with the observation that the users of the method must possess considerable conceptual, abstract and philosophical skills. Moreover, SSM requires that its users exercise a high degree of interpersonal skill if political pitfalls are to be avoided, i.e. great care must be taken in selecting the content and timing of

a contribution to the debate. An earlier version of SSM includes the Formal Systems Model as the technique for structuring the systems design process. This was to be replaced in the later version by applying the CATWOE criteria to establish root definitions:

C - 'customers': the victims or beneficiaries of T

A - 'actors': those who would do T

T - 'transformation process': the conversion of input to output

W - 'Weltanschauung': the world view which makes this T meaningful in context

O - 'owner(s)': those who could stop T

E - 'environmental constraints': elements outside the system which it takes as given

From these declarations, an activity is constructed to support each root definition. While SSM may provide some clarity to 'ill-structured' situations (although the process itself seems precarious and highly contingent), it most certainly does not provide the basis for constructing corresponding systems. As Jayaratna notes, there is no distinction between the logical and physical dimensions of a system; it will be argued later that this is an essential discipline for any credible method. Moreover, to view a 'transformation process' as the conversion of input to output is an untenably simplistic interpretation: there are no constraints imposed on inputs and outputs, there is no concept of what constitutes a conversion, and there is no recognition that systems can be sequenced by allowing the output for one process to form the input to another process. With these limitations, it is highly unlikely that SSM is capable of transforming a situation. Rather the epistemological thrust of SSM emphasises the importance of debate and discussion as a means of empowering users to achieve greater levels of cognitive activity. However, while it remains unclear how intellectual reasoning and self-learning can be transformed into decisive action, the contribution of SSM to the knowledge creation cycle is dubious and it is therefore not considered any further in the thesis.

- Jayaratna's study of hard system methods merely examines the Structured Analysis and Systems Specification (SASS) method (DeMarco, 1979). Jayaratna comments that while it is very useful and practical for describing and designing computer-based information systems, a lack of conceptual foundations limits its utility (a fact that is readily acknowledged by DeMarco). Moreover, DeMarco is exhorted to include consideration of user requirements and the political process in any new version of SASS.

While Jayaratna's denouement cannot be challenged (the weaknesses of early structured methods have been well understood by the industry for some years), the selection of SASS as a representative of structured methods is somewhat disingenuous. Indeed, structured methods have evolved (and continue to evolve) at a prodigious rate and now address many of the concerns raised by Jayaratna. The basis of these comments no longer reflects the current state or concerns of structured methods. The remainder of this chapter is dedicated to evaluating the development of structured methods and the extent to which they have contributed to the knowledge creation cycle.

For the purposes of the thesis, it is convenient to classify analysis methods into three broad categories:

- the Structured Analysis methods
- the Information Engineering method
- the object-oriented (OO) school of methods.

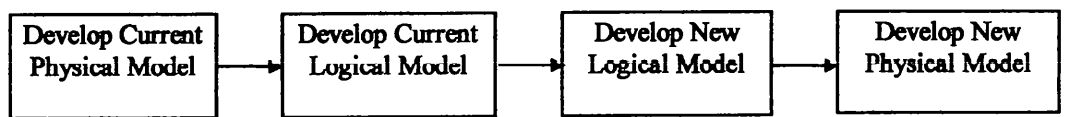
## 5.4 Structured Analysis

Sommerville (1989) suggests there are two dominant streams to Structured Analysis:

- top-down structured design

- data-driven design.

Popular accounts of top-down structured design may be found in the work of DeMarco (1978), Yourdon and Constantine (1979), Gane and Sarson (1979), and Page-Jones (1988) that derive generally from Dahl, Dijkstra and Hoare (1972). Top-down structured design concentrates almost exclusively on modelling systems; the principal modelling tools are the function decomposition diagram and the data flow diagram. Simplistic entity-relationship models are developed to support the systems. Top-down structured design is based traditionally on the production of four system models:



**Figure 5-1 The top-down structured design lifecycle**

While top-down structured design appears to be an eminently reasonable way to proceed, the two principal techniques are critically flawed. The construction of function decomposition diagrams present four difficulties:

- when does a function decomposition begin, i.e. how does one determine its root node?
- when does a function decomposition end, i.e. how does one determine each leaf node?
- what are the criteria for including a node on a branch?
- what are the criteria for excluding a node from a branch?

While these questions remain unanswered the structure and content of the traditional function decomposition should be viewed with caution.



Data flow diagrams also present four difficulties:

- the analyst is asked to accept the metaphor that data flows between processes; as this may not actually happen in the real world, the metaphor may be placed under strain and impose cognitive barriers
- where it is accepted that data flows, all flows between processes must be shown; where flows are inherited through multiple levels of decomposition, a lower level diagram can become overwhelmed with data flows probably concealing more than is revealed - the flows become little more than noise
- the mere existence of data stores invites design decisions to be introduced to logical models
- a similar problem exists with external entities; their instantiation is virtually unconstrained and consequently there is no restriction on them being used to introduce physical content.

Another problem common to both functional decomposition diagrams and data flow diagrams is that there is little guidance on precisely what type of activity is included in each model. It was not until McMenamin and Palmer introduced the concepts of an essential model and an implementation model that these deficiencies became apparent (McMenamin and Palmer, 1984). The essential model is defined as '*what the system must do in order to satisfy the user's requirements, with as little as possible (and ideally nothing) said about how the system will be implemented*'. As Yourdon notes, this assumes perfect technology is available at zero cost (Yourdon, 1989). The essential model consists of an environmental model defining the boundary between a system and its environment and a behavioural model describing the behaviour required for the system to interact successfully with its environment. The implementation model is a physical model of the existing system that implements the environmental model. How the implementation model is populated is regarded as a matter of individual interpretation of what constitutes the more critical (or important) processes in the model. The imperative is to 'avoid modelling the user's current system *if at all possible*', and commence modelling the essential model '*as quickly as possible*'. To extract the logical essence of an implementation model to construct an essential model is undoubtedly considered to be an option of last resort. Clearly, what is being attempted here is to emphasise the essential differences between logical (the *what*) and physical (the *how*) models, and the necessity to ensure

that the two models remain separate. Moreover, the constitution of a logical model is becoming consolidated and better understood.

However, guidance on how exactly to populate a logical model remains tentative. The inclusion (or exclusion) of activities in *logical* data flow diagrams is based on familiar heuristics; there is no indication that business processes are the only type of activity that may populate a data flow diagram. This is not surprising, as the process is not recognised as the pre-eminent activity. The issue is discussed in more detail in the chapter addressing business processes.

Data-driven design differs from top-down structured design in that it has a theoretical, rather than heuristic, foundation. Böhm and Jacopini (1966) defined the original rules for structured programming. Data-driven design originated with the Warnier-Orr method following research into information domain analysis by Warnier (1974; 1981) concluded that software structures could be derived from the sequence, selection and iteration of tasks executed on data structures. Orr extended the Warnier notation to represent ~~information-items-and-the-associated-processing-requirements-and-so-describes-data-structured-systems~~ development (Orr, 1977; 1981). Jackson (1975; 1983) further developed the concepts by integrating them more closely with system and program design. The theory of database design was developed by the work of Codd (1970) and Date (1983) who applied the relational model to specify logical data structures as sets of associated tables. Chen introduced an alternative approach to data modelling with the concept of binary relational modelling which represents attribute values as separate simple relations associated with the entity they describe (Chen, 1976). The major advance represented by data modelling is that data items may be defined independently of any considerations of physical database organisation. The distinction between logical and physical representations of data is established.

However, even with its theoretical credentials, data-driven design presents two significant problems: how the existence of data is to be discovered and defined precisely? Those texts that do address this issue rely simply on some grammatical parsing of a domain definition, interview transcript or a requirements

statement. Such documents rarely exist in practice and are nearly always inadequate for this purpose. The 'pragmatists' will doubtless plead that heuristics and inference will reveal most domain data; however, for complex domains there is also a point beyond which heuristics and inference are incapable of proceeding. Alternatively, existing data schema may be reverse engineered to construct a normalised information structure for a system domain. However, to simply create a set of tables that have no embedded structure fails to capture the semantics of a system domain, i.e. any correspondence between the normalised tables and entities populating the 'real world' of the system domain is merely incidental. Data-driven methods are concerned with syntactic analysis and not the semantics of a domain. To conduct domain analysis from a data-driven perspective would seem to be as equally dubious as adopting a process-driven approach.

To conclude the examination of Structured Analysis methods, it remains to evaluate them against the criteria established at the beginning of the chapter. With respect to system taxonomy, Structured Analysis has the capability only to address simple systems that possess minimal structure and are largely static. Although there is some recognition of hierarchy, the resultant structures are so arbitrary that they must be of dubious value. Structured Analysis does, however, make a limited contribution to the knowledge creation cycle. The use of models reveals to the domain expert the possibilities of abstract reflection and expression. The abstraction and codification conventions are, however, too primitive to support anything more than superficial externalization before cognitive barriers are erected. Accordingly, the prospects for internalisation are poor. It is doubtful, therefore whether much progress would be made on Boisot's SLC (Boisot, 1995). Finally, the models would have limited utility and their superficiality would compromise the scarcity of the knowledge embedded in the models.

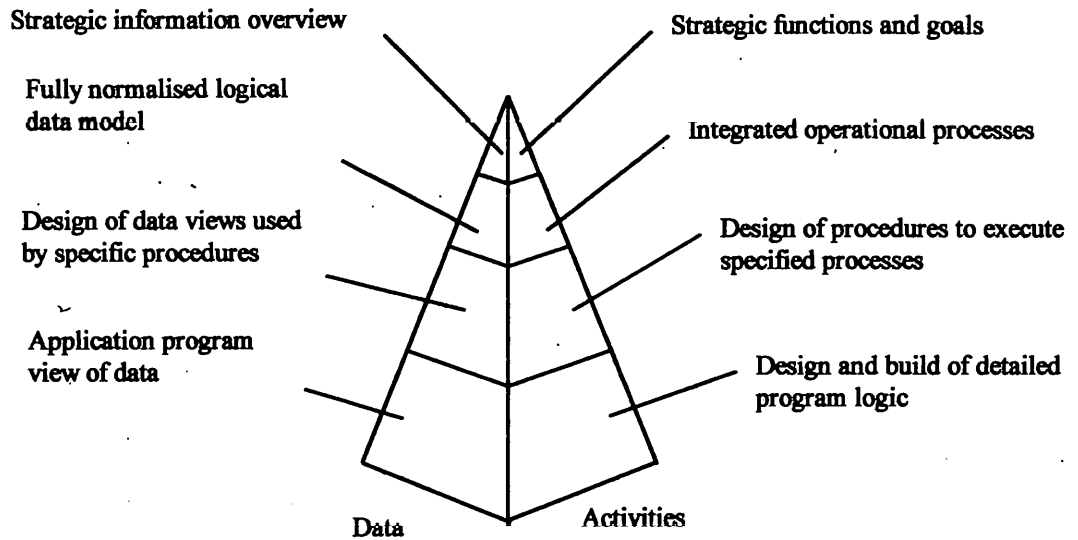
## **5.5 Information Engineering**

Information Engineering (IE) was popularised by James Martin, but has its origins in concepts developed at IBM in the 1970s where Martin and Clive Finkelstein attempted to integrate Information Systems (IS)

and Information Technology (IT) with strategic business planning (Martin & Finkelstein, 1981, Martin 1989)), and from the CACI method developed in the 1970s (Palmer and Rock-Evans, 1981; Macdonald and Palmer, 1982).

IE seeks to deliver a corporate solution integrating all aspects of the enterprise. To achieve this level of integration, information is viewed as a corporate resource and thus differs significantly in this respect from conventional methods that view information as the province of individual applications. Managing information as a corporate resource necessitates a definition of the information based on high-level business plans, policies and strategies from which various architectures may be constructed to express a comprehensive and cohesive statement of corporate information needs. The strategic business architectures are subject to detailed analysis usually focusing on specific business areas, from which logical models provide a definition of functional and non-functional requirements, i.e. the IS problem domain architecture is defined. The IS requirements are then clustered into logically cohesive partitions and subjected to the familiar design and construction activities. By observing the distinction between the problem and solution domains and adopting a holistic perspective, proponents of IE claim to produce fully integrated business, IS and IT architectures delivering software solutions that verifiably support strategic business requirements.

The pyramid has become linked inextricably with IE as a means of expressing its conceptual structure and foundations. As the method has evolved, the content and structure of the pyramid have changed accordingly. However, most IE practitioners would recognise the pyramid represented in Figure 5.2 as capturing the essence of the method. Later versions of the pyramid have included a third side to represent a technology architecture.



**Figure 5-2 The IE Pyramid (Martin, 1986)**

The use of a pyramid as a metaphor for IE implies a 'top-down' approach to the system development lifecycle, i.e. all architectures flow from the business strategy. This departs from Structured Analysis methods that subscribe to what is commonly referred to as the 'bottom-up' approach where an existing physical system rather than a business strategy is considered to be the point of origin.

From the IE perspective, it might be argued that the 'top-down' approach exposes the following weaknesses in Structured Analysis methods:

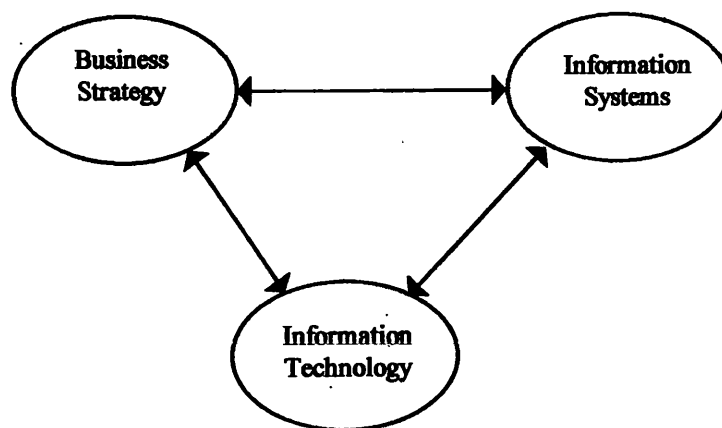
- the full software engineering life cycle is not addressed by conventional methods and those phases that are covered provide inadequate support for modeling activities (functions and processes)
- the emphasis on existing procedures undermines the opportunity to re-engineer business processes
- verifiable support for strategic information needs is problematic as domains are confined to an application area

- integration between applications is poorly supported
- existing systems contain inadequate and often extraneous documentation.

However, IE is also not without its critics. The scope and ambition of the method gives rise to the greatest comment. Although IE 'experts' claim impressive achievements, there is a persistent anxiety that attempting to analyse an entire enterprise is a far too complex and ambitious an undertaking which will lead inevitably to never-ending projects and the spectre of 'paralysis by analysis'. Moreover, IE is extremely dependent on and sensitive to the sustained collaboration of the business community that is rarely forthcoming when projects founder on the rocks of methodological rigour.

Notwithstanding the criticisms, the method has much to offer because great care has been taken to define precisely the tasks, techniques and deliverables required for each stage of the system development lifecycle. A summary is now included of those features salient to the thesis.

IE is designed to provide a basis for developing IT solutions which support corporate business objectives by integrating business strategy planning with IS and IT planning and development.



**Figure 5-3 The triumvirate of IE architectures**

For the IE method to be effective, the business strategy architecture should be expressed as a fully articulated, quantified and prioritised business plan. From this platform, an IS architecture is constructed comprising an information strategy plan and a logical business model. Only when these are in place, is it possible to proceed to the more familiar terrain of software design, construction, testing and implementation. There is also a degree of recognition for the possibility of feedback, where a change in one architecture may result in changes to the others. However, it should be emphasised that the trajectory is unerringly top-down.

The IE system development process commences with the Information Strategy Planning (ISP) which seeks to provide a prioritised definition of the IS/IT requirements necessary to achieve strategic business objectives. A strategic business plan is required detailing strategic objectives, goals, CSFs (Critical Success Factors (Rockart, 1979)), performance measures, and problems; these strategic objects are then mapped onto infrastructure components such as products, processes, organisation units and locations. The level of support provided by the existing information management organisation and IT resource inventory is evaluated during the ISP.

From this analysis, a strategic business architecture is constructed and mapped onto the IS and IT architectures to associate all dimensions of the IE pyramid. By defining these holistic associations, it is possible ultimately to evaluate the level of support provided by various elements of IS and IT for each strategic requirement, i.e. a gap analysis is conducted to determine the discrepancy between the actual and required level of support for each strategic requirement. The gap analysis quantifies where the need is greatest and the support is least, thus providing the intelligence from which to formulate a prioritised migration strategy. The strategic priorities are assigned to the dozen or so business areas comprising the entire organisation.

In IE parlance, detailed analysis is referred to as 'Business Area Analysis' (BAA), thus reflecting the holistic ethos of the method. For someone immersed in IE, it is counter-intuitive to confine analysis to an application area; the only viable option is to analyse (at least) an entire business area. The objective of a BAA project is to express functional requirements in terms of a logical business model; in particular, the task is concerned with defining the object types populating the model, together with appropriate descriptions of the associations, structures and corresponding business rules. Development of three orthogonal views (i.e. models) is required to fully define the logical business model, i.e.:

- the static view, defined from entity analysis
- the functional view, defined from process analysis
- the dynamic view, defined from interaction analysis.

At the level of detailed analysis, IE achieved a significant advance by defining a new approach to analysing business activities. The innovations in this area are profound and denote a more significant departure from Structured Analysis than was probably appreciated at the time. Although IE betrayed a decidedly data-driven preference at the outset, it evolved rapidly into a method that concentrated on processes as the primary focus of analysis. The first contribution is to study more closely the nature of business activity. From this study three types of business activity are identified:

- at the most generic level there is the business function which simply represents some loosely cohesive cluster of ongoing activities, e.g. Product Ordering, Sales Planning, Financial Planning, Cash Management
- more specifically there are business processes executed within a finite time and achieving a business change, e.g. Place Order, Execute Deal, Settle Trade, Recruit Employee
- finally there are business procedures designed to implement processes.



These definitions are significant in that they alert the analyst to the type of activity that should be included in (or excluded from) from a specific model.

The notion of data flowing between processes is replaced with the concept of dependencies between processes. Modelling processes becomes immediately more intuitive. It is no longer necessary to make the conceptual 'leap of faith' that data flows between processes; the fact that one activity is dependent on another has a much stronger affinity with reality. A dependency between two processes is represented by a change of entity state, and there is thus a bonding between data and processes that could never be possible with data flow diagrams. Moreover, activity models may be stripped of the noise generated by a superfluity of meaningless data flows. Another compelling advantage of modelling dependencies is that the invocation of one process is rarely dependent on the completion of more than two or three other processes; greater immediacy and clarity of expression is thus achieved. Process dependency diagrams thus impose the constraint that entity analysis cannot be completed until process analysis is complete; the implication here is clear: the heuristic approach cannot be relied upon to complete an analysis exercise, for this it is necessary to ensure that processes and data are fully associated by enforcing the rule that a process changes the state of data.

The BAA phase concludes by associating what are referred to as non-functional requirements with the processes, data, organisation units and locations, and by preparing generally for the Business System Design phase. For the purposes of the thesis, IE is only of marginal interest beyond this point.

As Structured Analysis considerably pre-dates the fully evolved IE, it is not unexpected to discern significant changes in scope, emphasis and sophistication. While Structured Analysis methods prepared many of the foundations on which subsequent methods and techniques were constructed, it is clear that when compared with the sheer comprehensiveness of IE, they are equipped only to confront simple systems. The more interesting evaluation of IE is therefore to examine how it is equipped to manage complex systems.

It is clear, even from this brief account, that hierarchies permeate every aspect of IE. Not only is the essential metaphor hierarchical, but also hierarchies pervade each layer of the IE pyramid. It may therefore be concluded that IE is predicated on the view that a system is a hierarchy. Moreover, it may also be inferred from the emphasis on corporate solutions that a holistic approach is used to analyse systems. The perception of systems is quite different from Structured Analysis where a system is viewed as some simple constellation of related objects. However, research into fields such as process design and improvement have established that corporate behaviour does not conform to some neat hierarchy. As Davenport reports, the subordination of process to function is problematic (Davenport, 1993). IE does not allow for processes to cross functional or organisational boundaries, but this is precisely how most processes are implemented in practice. The reliance on the hierarchy thus militates against the use of IE as an effective agent of change and innovation.

While the importance of a strategic direction is acknowledged within IE, adherence to the top-down trajectory conspires to inhibit the emergence of strategic content from tacit knowledge. Consequently, a potent source of requirements is vulnerable to neglect. However, a series of alternative procedures is now available within the IE framework with the potential to penetrate tacit knowledge although this is not their stated aim; the procedures include:

- the single-iteration lifecycle
- the multi-iteration lifecycle
- the timebox lifecycle
- the evolutionary lifecycle
- the exploratory lifecycle
- the quick-results lifecycle
- the decision-support lifecycle
- the expert-system lifecycle.

The holistic aspirations informing IE allow it to address a wider range than those tackled sensibly by Structured Analysis. From the earlier discourse on the origin and nature of requirements, it can be argued that an holistic approach to analysis might well reveal knowledge assets and requirements that would otherwise remain concealed. The importance placed on the Process Dependency Diagram seeks to establish a conceptual integrity throughout the analysis phase and to also influence the structure of the adjacent phases. However, in common with Structured Analysis, the codification conventions impose a compromise on the representation of domain concepts, e.g. concepts do not necessarily reduce to the two-dimensional expression required by entity-relationship diagrams.

The final verdict on IE is mixed. The holistic foundations of IE allow the method to tackle systems of greater complexity than the Structured Analysis methods. The metaphorical pyramid is only relevant to those organisations (or parts of organisations) that subscribe to unitary and mechanistic hierarchies. Where an organisation departs from the top-down emphasis of the pyramid, IE has little to offer. So where does this place IE as an analysis method to support knowledge creation? The abstraction and codification conventions support some form of sustained enquiry within a top-down paradigm. However, even within this paradigm, there is a loss of semantic content in the expression of knowledge assets and requirements for change. The externalization and internalization processes are therefore under threat. Although IE may yield more knowledge assets and requirements than the Structured Analysis approach, it shares the problem that the utility and scarcity of these assets is compromised.

## **5.6 Object Orientation**

The apparent advances heralded by the maturation of OO have also led to extravagant and unsubstantiated claims being made on behalf of OO analysis methods. The probable reasons for this are threefold:

- The necessity to respond effectively to competitive turbulence is now a fact of corporate life. If subtle environmental shifts remain undetected or neglected, their consequences may escalate rapidly beyond the capacity of the organisation to seize an opportunity or deflect a threat. It is argued that those organisations better able to adapt are more likely to survive the onslaught of unpredictable change.
- Organisations cannot adapt if their IT systems are predicated on some assumption of stability based on a unitary hierarchy, and a prescriptive, 'top-down' approach to strategic management. While such considerations are beyond the remit of Structured Analysis, the philosophy of IE is based exclusively on hierarchies and top-down system engineering. Far from contributing to corporate adaptability, IT solutions based on these methods appear more likely to be obstructive.
- Tapscott and Caston report that IT provides the means by which an organisation may manage the phalanx of paradoxes presented by the combinatorial impact of change, complexity and uncertainty (Tapscott & Caston, 1993). IT is identified as the panacea and yet remains incapable of making a decisive contribution (yet another paradox). These trends, which at this time show no sign of abating, have contrived increasingly to marginalise non-OO methods. The emergence of OO as an analysis method is therefore timely.

#### **5.6.1 The Fundamentals of OO**

Graham (1991; 1994) provides a detailed account of the background to OO development techniques. It is reported that the conceptual origins can be traced back to the development of the Simula language in 1967, with the term 'object-oriented' first used to describe the Smalltalk language developed by the Xerox Palo Alto Research Centre (PARC) in the early 1970s. The Flex machine concept emerged from doctoral research undertaken by Alan Kay; the concept anticipated much of the design of the modern PC workstation, i.e. a ubiquitous, small machine capable of allowing the non-specialist user to manipulate

information in multiple formats. With the Smalltalk language embracing features of both Simula and LISP, PARC were able to develop an early version the Flex machine referred to as Dynabook.

During the 1980s, OO concepts were used by Xerox and Apple to pioneer development of graphical user interfaces (GUIs) based on the WIMP (Windows, Icons, Mice and Pointers) paradigm. Artificial Intelligence languages were also extended by OO constructs to enrich knowledge representation by utilising concepts such as: semantic networks, frames, slots and demons. As Smalltalk focused primarily on GUI applications, Graham suggests that languages such as Eiffel and C++ appeared in the late 1980s to address a wider range of applications.

While OO techniques were confined largely to programming disciplines during the 1980s, they were extended progressively in the 1990s to become transformed into mature design and analysis methods. The application of OO techniques liberated system engineers from many of the limitations imposed by the *waterfall* model; rather than absorb the deterministic constraints of the 'top-down' approach, it became possible to view system development as an evolutionary series of incremental iterations.

Naively, objects (or classes) combine data and processes, i.e. they manifest structure and behaviour. From this modest declaration, the concepts that provide the potency of OO may be defined as follows:

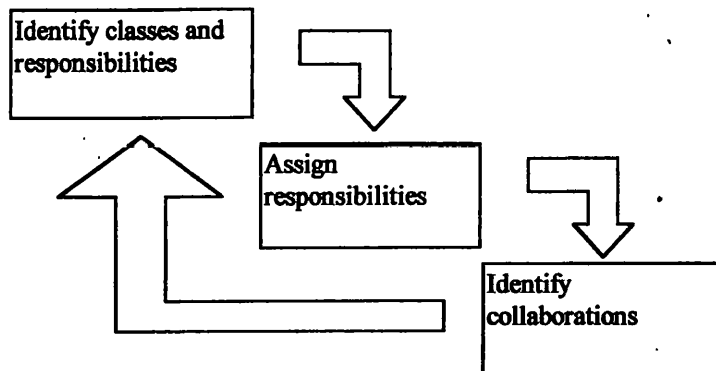
- Abstraction enables all essential features of a class to be represented while incidental detail is used to form associated classes. Abstraction enables concepts to be defined according to a scale of genericity.
- Encapsulation is concerned with ensuring that the data structure defining a class is only accessible through operations assigned to the interface of the class. The terms encapsulation and information hiding are used interchangeably.

- Inheritance provides the means by which stereotypical structures may be exploited to capture the semantic structure of a domain concept. Quite simply, inheritance allows a specialised class to inherit the structure or behaviour of a more generic class within the same classification hierarchy. However, the inheritance requires that a class is defined fully by its structure and behaviour, and that a specific class inherits all the structure and behaviour of a generic class.
- Finally, class behaviour may only be invoked by messages passed from other objects; this includes the concept of self-recursion where a class may send a message to itself.

From these definitions, Taylor notes the obvious analogy with cellular structures that are organised hierarchically as cells, organs, systems and organisms (Taylor, 1992). This analogy is at the core of the interest in OO as an analysis method; at last an analysis method is available with the potential to deliver systems that mimic the behaviour of organisms that survive by adapting successfully to change. The remainder of this chapter evaluates how the major commercial OO methods have fulfilled this potential.

### **5.6.2 The Responsibility-Driven Method (RDM)**

RDM represents an early attempt to apply OO concepts to domain analysis (Wirfs-Brock *et al*, 1990). RDM comprises an exploratory phase followed by the analysis phase. The exploratory phase is an adaptation of the CRC (Class Responsibility Collaboration) technique (Beck & Cunningham, 1989), which assumes an anthropomorphic perspective requiring actors to enact scenarios by performing the roles of prospective objects to identify classes, responsibilities and collaborations. Details are recorded on index cards (the limited space available on the card is designed to enforce the appropriate level of abstraction).



**Figure 5-4 CRC: Process Overview (Coleman et al, 1994)**

The exploratory phase is concerned with identifying candidate classes and inferring possible responsibilities and collaborations for each class. The approach to identifying classes, responsibilities and collaborations is largely heuristic and based primarily on parsing domain declarations for noun phrases and implicit nouns to expose possible classes. Responsibility assignment is achieved by attempting to evenly distribute responsibilities across the domain. Class collaborations are identified by examining each responsibility to determine those classes which must collaborate from either a client or server perspective. 'Is-part-of', 'has-knowledge-of' and 'depends-upon' relationships between classes may also yield collaborations. Classes that do not participate in any collaboration are discarded.

Class Name	
Super-classes	
Sub-classes	
Responsibilities	Collaborations

**Figure 5-5 CRC: Class Card (Coleman et al, 1994)**

For the analysis phase, Wirfs-Brock seeks to add structure to the preliminary design produced during the exploratory phase. Attention is focused, in particular, on defining inheritance hierarchies and

encapsulating functionality by clustering cohesive groups of classes into subsystems. Finally, guidelines are provided for developing class protocols from which specifications are produced for classes, subsystems and contracts (a collaboration between two classes performing client and server roles).

RDM is an empirical method predicated almost entirely on heuristics; not surprisingly, RDM contains intrinsic limitations. Bellin and Simone acknowledge that CRC requires intervention from more formal OO methods (Bellin and Simone, 1997). The limitations are exposed most crucially, however, in the definition of a subsystem as a 'set of contracts providing a clearly delimited unit of functionality'. The weakness here is that no definition is given of a clearly delimited unit of functionality', i.e. what is a unit of functionality and when is it clearly delimited? Furthermore, Wirfs-Brock attempts to distribute intelligence evenly throughout a system; intelligence is considered to be too concentrated when a subsystem or class supports too many contracts. Where too many contracts are supported, the internal structure of the subsystem is examined to determine if embedded subsystems can be identified and contracts distributed among existing subsystems. The lack of definition and discrimination in describing systems and their constituents leads inexorably to many of the difficulties identified previously for Structured Analysis.

### **5.6.3 The Coad & Yourdon method**

The Coad-Yourdon method first surfaced in 1989 (Coad, 1989) and was described later in greater detail (Coad & Yourdon, 1990; 1991). The speedy revision reflected both the perfunctory content of the original volume and the more substantive material published elsewhere.

Coad & Yourdon describe a multi-layered model to capture requirements and organise analysis and design activities:



- the class-&-object layer
- the structure layer
- the subject layer
- the attribute layer
- the service layer.

An attempt is made with the 'Class-&-Object Layer' to clarify the terms 'class' and 'object' by providing formal definitions supported by enhanced diagramming notations and familiar discovery techniques. The discussion commences by quoting rather selectively from Meyer (1988) to assert that objects are not 'just there for the picking' and proceeds to provide definitions of classes and objects. While Meyer recommends the construction of an *operational model* with objects identified from an analysis of the structure and *behaviour* of a system, Coad & Yourdon exhort the analyst to:

- observe
- actively listen
- check previous OOA results from similar problem domains
- 'read, read, read the "requesting document" '.

Thereafter prototyping techniques are suggested to secure *descriptive* expressions of requirements while always endeavouring to absorb the system terminology and semantics. From these sources, Coad & Yourdon propose that class-&-object occurrences are identified from:

- generalisation-specialisation structures
- external system interactions
- physical devices (at some appropriate level of abstraction)
- things or events remembered
- roles played

- operational procedures
- organisational units.

The approaches to all other layers reveal a similar predilection for some nonspecific blend of heuristics and empiricism which is conveyed in a style likely to alienate all but the novice analyst. The method is illustrated with a series of simple examples omitting any attempt at verification or validation, emphasising rather their preference for establishing intuitive correctness. In contrast, Meyer provides a beacon for extolling the virtues of greater formality.

Coad and Yourdon aspire to impose some structure on proceedings without sacrificing the holy grail of 'pragmatism'. While the structure implied from the layers is welcome, population of the layers remains problematic. Elicitation of the required details is based loosely on a semantic appreciation of a domain; this approach is only effective where the domain is familiar (where residual understanding is available) and does not contain undue complexity. As software engineering is required increasingly to address applications characterised by accelerated change (and therefore reduced familiarity) and growing complexity, the utility and general applicability of the Coad-Yourdon method must be in some doubt. Finally, and critically, behaviour is completely subordinated to structure; the analyst is expected to discover behaviour by examining object states and inferring the existence of object methods. The contribution of the business process in discovering and authenticating the structure and behaviour of a business object is not recognised, i.e. the primary agent for discovering objects is completely neglected.

#### **5.6.4 The Object Modelling Technique (OMT) method**

OMT (Rumbaugh *et al*, 91) consists of constructing a logical application domain model followed by a design phase where implementation details are added to the model; the method is divided into the following phases:

- *Analysis:* Application requirements are analysed to define a logical model of the problem domain in terms of application objects, functionality and behaviour. The expression 'logical' conveys the constraint that the model represents only *what* is required of the desired system (i.e. the functional requirements) and therefore does not contain any physical implementation detail. Deliverables from this phase are: a problem statement, an object model, a dynamic model and a functional model.
- *System Design:* The target system is organised into sub-systems based on an assessment of the problem domain model and high-level decisions concerning the proposed technical architecture. System design is therefore concerned with designing an environment that will satisfy the non-functional requirements. Deliverables from this phase are: a high level design strategy, a system architecture definition.
- *Object Design:* The second design phase is concerned with augmenting the logical model to support the design decisions formulated during system design; physical domain data structures and algorithms are added to the models to satisfy the non-functional requirements. Rumbaugh notes that although the logical and physical domains express different conceptual perspectives, they can both be defined using common OO concepts and notations. Deliverables from this phase are: a detailed object model, a detailed dynamic model, a detailed functional model.
- *Implementation:* Finally, the models developed during object design are translated into software components and implemented on selected technical platforms. Emphasis is placed on the necessity to adopt good software engineering practices to achieve design traceability, flexibility and extensibility (the primary benefits of OO technology). The deliverable from this phase is an implemented system developed using OO languages, non-OO languages and (optionally) some form of dbms.

However, for the purposes of the thesis, the evaluation focuses primarily on the analysis phase that, coincidentally, also constitutes the majority of the OMT content.

The purpose of analysis is to develop three orthogonal logical models of an application domain (the application domain model):

- the *object model* representing the structural definitions expressed in terms of complex data types, associations and operations
- the *dynamic model* representing the behavioural definitions expressed in terms of control, sequencing of operations and object interaction
- the *functional model* representing the value transformation definitions expressed in terms of functions, function dependencies, mappings and constraints.

Object models are populated with: objects and classes, associations, aggregation hierarchies; and generalisation hierarchies supported by inheritance mechanisms. Rumbaugh suggests that model content is a 'matter of judgment' and intuited from an appreciation of the application domain; further, an object may be conceptual, abstract or a 'thing with crisp boundaries' and have meaning for an application. Familiar heuristics are provided to eliminate spurious classes, attributes and associations; Rumbaugh suggests generalisation hierarchies are usually apparent from noun phrases containing adjectives, e.g. *fixed* menu. Access paths are verified by tracing routes through the model to ensure 'sensible results' are provided to questions concerning optionality and cardinality. Object operations are derived primarily from dynamic and functional models but Rumbaugh allows for 'shopping list' operations, as defined by Meyer, where operations are not contained within the scope of the application domain, but essential to the management of the object (Meyer, 88).

Dynamic models express the permitted ordering of events, states and operations within an object model; the models follow the statechart notations (Harel, 87). Construction of a dynamic model commences with

preparation of scenario descriptions defining the interaction sequences of typical and exceptional dialogues within the application domain. From each scenario definition, events between objects are identified and translated into event traces enabling the construction of event flow and, ultimately, state diagrams.

Where the object model represents the structure of objects and their associated predicates, and dynamic models show the order in which operations may be performed, Rumbaugh describes functional models as defining the computation and derivation of data values without any reference to 'how, when, or why the values are computed'. Data flow diagrams (DFDs) are used to express functionality. Rumbaugh suggests that functional modelling commences after object and dynamic modelling, by identifying input and output values describing events between the application domain and the external environment. Conventional DFD techniques are then applied to describe all nontrivial processes and associated dependencies in the application domain; when the DFD 'has been refined enough'; the procedural logic for each process is described in natural language.

Rumbaugh's OMT method represents a significant advance by providing three orthogonal views with which to define an application domain: the static view, the dynamic view and the functional view. Progress is made establishing and verifying the contents of the static view; it is no longer necessary to rely on individual judgement and perceived meaning within the application domain context. Through the use of dynamic and functional modelling, Rumbaugh acknowledges the association between activity modelling and constructing object models; however, the association is not conclusive. Again, the initial discovery of business objects is dependent on grammatical parsing of some domain definition. Activity modelling is introduced at some indeterminate point to discover the object operations; presumably, a side effect of activity modelling is that the object model may also be progressed from its heuristic foundations, although this appears to be overlooked by Rumbaugh. The verdict on OMT is inconclusive; although an advance has been achieved, it is undermined by conceptual weaknesses.

### **5.6.5 The Ptech method**

Ptech (Process Technology) aspires to venture beyond software engineering, by presenting a series of OO techniques with greater emphasis on conceptual considerations rather than physical expressions to address the planning, analysis and design of systems for 'people, machines or computers'. Class theory, logic and process philosophy form the conceptual basis of Ptech, providing an integrated Process Engineering environment.

Ptech (Edwards, 1984) comprises concepts, techniques and methods to produce a 'process prototype specifying a computer system'. Concepts include: objects, functions, events, processes and trigger rules. The following techniques are based on formal mathematics and seek to provide abstractions reducing complexity and improving design re-usability:

- class calculus - uses Relational Calculus and Event Calculus to adapt existing class definitions to define new classes
- functional calculus - mappings between classes incorporating inference and constraint rules are used to adapt existing function definitions to define new functions
- process calculus - adapts process descriptions to describe new processes.

Ptech is designed to integrate the static and dynamic views of an activity, i.e. 'the object and changes which comprise an activity', with the provision of three methods:

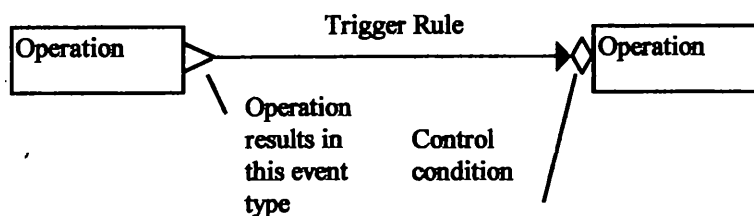
- event analysis - identifying event types representing the successful completion of an invoked operation resulting in a change of object state
- concept analysis - grouping object types into associated classification hierarchies
- activity analysis - specifying process functions which transform the products consumed by a process (the resources) into the product generated by the process.

The Ptech method focusses on the way people perceive reality, i.e. the *concepts* that have to be managed within an application domain. Concepts thus provide the initial source of candidate objects supporting a common understanding of the domain. Concepts are defined further as units of knowledge permitting the introduction of a concept triad where a concept definition includes intension (definition of recognition tests) and extension (those objects satisfying the definition). Therefore an *object* becomes 'anything to which a concept applies' and a *set* is the collection of objects satisfying the concept intension, i.e. the extension of the concept.

The application of abstraction, generalisation and composition mechanisms on concepts is defined as an approach to managing complexity. Abstraction is viewed as an iterative process of removing distinctions between objects to reveal the commonalities and identify new concepts; the approach is derived largely from Langer (Langer, 1967). The Ptech method makes provision for an object type to be defined in terms of partitions of disjoint subtypes. Object partitions offer the useful facility of relaxing exclusive subtyping when describing an object, by selecting subtypes from multiple partitions (providing for inclusive and overlapping definitions), while also preserving the exclusivity of subtypes within individual partitions. An object may have multiple supertypes and multiple subtypes. Ptech also provides for incomplete partitions (i.e. a partial list of subtypes) that may be used to define individual application views. Hierarchies are viewed as providing depth to an object definition. The ability to define multi-level partitions provides width to subset schemes, enabling effective bridging between macro- and micro-views of an object. Partitioning at multiple levels also enables improved expression of underlying meaning and reduction of ambiguity; the process is referred to as strengthening (Borgida, 1984) or sharpening an object structure. Associations between object types may be expressed as relations or functions. Relations are used when object types are viewed as tuples, while functions map objects of one type (the domain) to a set of objects of the same or another type (the range).

Where the structural definition describes how to recognise objects and their associations, behaviour is viewed as a description of how that recognition may vary over time. At a superficial level, the view of system behaviour adopted by Ptech is virtually identical to that used by other OO methods, i.e. system behaviour is described by a network of state changes with respect to an object structure, i.e. changes in the collection of object types applying to an object. However, with Event Schemas, a more rigorous approach to modelling system behaviour is introduced through the Ptech method. Unfortunately, the greater rigour comes at a price; Ptech method is the most complex method under consideration in the thesis. The complexity is introduced with the rigorous constraints that must be observed before an object can change state. Event schemas describe the sequence of events needed to achieve a goal within a system. The implications of that simple proposition are quite profound:

- most crucially, a system must have a single goal; if there are multiple goals, there must be an equal number of systems
- a system achieves its goal through the invocation of a series of operations in a prescribed sequence(s)
- each operation is triggered by events and is also terminated by an event (which results eventually in the realisation of the system goal)
- events always result in the state transition of an object within the system domain
- state transitions are 'guarded' (with control conditions) such that they may only occur when preceding transitions have occurred.



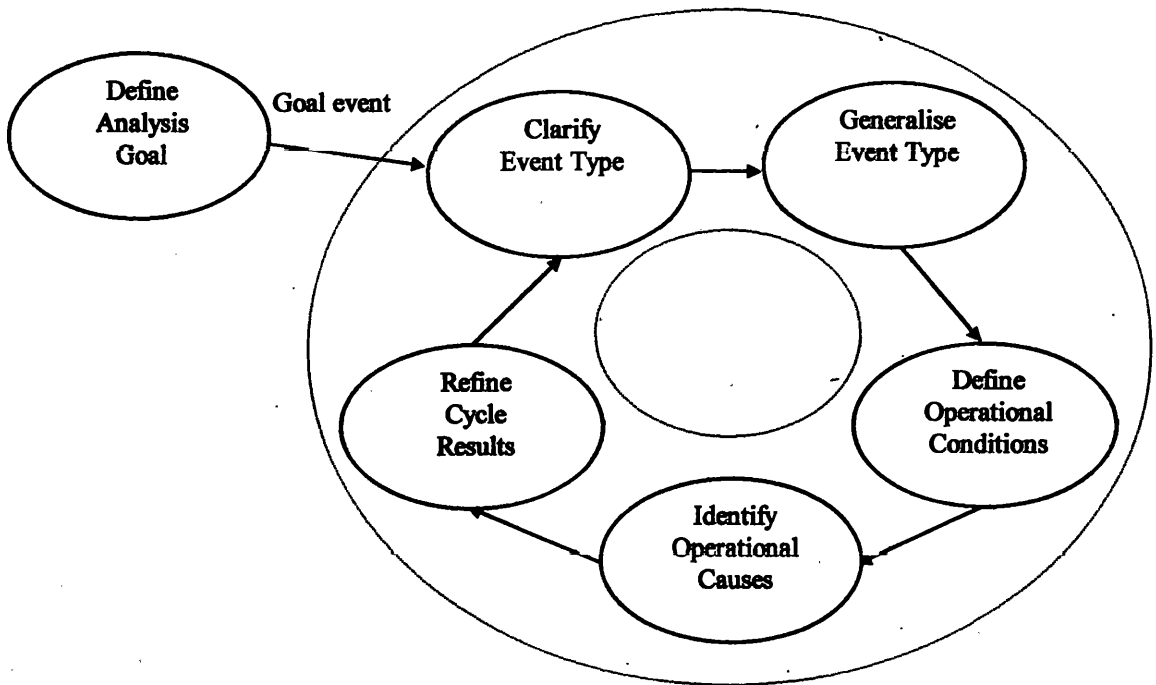
**Figure 5-6 Event schema components**



Event schemas are constructed by determining a preliminary realm of interest (the application domain focus) from which to define a universe of objects. A realm of interest is viewed as a method of operation, it is necessary to identify the starting and goal event types to define the scope and understand the purpose of the method. Once the goal event type is identified, the analysis iterations may commence. Before describing the event analysis method, it is worth noting that the Ptech method imposes the constraint that activities, at all levels of abstraction (e.g. systems, processes, operations), must have a single purpose, i.e. a single terminating event. An activity either succeeds by achieving its goal event, or fails by terminating without achieving its goal event; there is no other outcome.

The Event Analysis lifecycle commences by defining the goal, and thus the goal event type (i.e. the terminating event type), for the activity under consideration. Once this is established, a somewhat counter-intuitive approach is adopted by working backwards in an iterative trajectory until all the events triggering the activity have been discovered. At this point, the event analysis is complete. The iterative trajectory consists of the following phases:

- clarify the event type (the terminating event at the first pass)
- generalise the event type
- define the operation conditions
- identify the operation causes
- refine the cycle results.



**Figure 5-7 The Ptech Event Analysis Method**

Clarifying the goal event type involves first assigning a classification to the event and specifying the associated event pre-states and post-states. Rigor is introduced by expressing formally the kind of state change and the associated object types.

Generalisation of the event type is the technique by which an appropriate level of abstraction is determined and any opportunities of integration are exploited if the generalised event type is specified elsewhere.

The control conditions for the operation resulting in the goal event type are considered next. An external process (outside the realm of interest) requires no further analysis, while the method for an internal process is expressed as another event schema.

Great stress is placed on identifying the causes for each operation, i.e. for each control condition, what events must occur for the control condition to be evaluated as 'true' and, from those events, what subset actually trigger the operation?

An iterative cycle concludes with a refinement of the event type definitions produced from the preceding stages. Event types are examined to determine whether trigger events can be expressed as subtypes of more generalised event types; or, conversely, whether goal events are too abstract and should be expressed as a set of specialised subtypes with discrete operations. Event schema refinement also requires the elimination of duplicate event types. All new triggering events are identified from the event schema, and the event analysis process applied to each event that is now viewed as a goal event. The process is repeated until only external processes remain and the starting event is reached, i.e. there are no more internal events.

Ptech may be evaluated by considering why it is set apart from the other OO methods. It has been demonstrated by Ptech that heuristics can be dispensed with when defining the structure and behaviour of a domain; indeed, as Graham reports, Edwards considers reliance on heuristics to be a 'naive' approach to analysis (Graham, 1991). The risks associated with the heuristic approach are obvious:

- material crucial to the domain definition may remain undetected
- the analysis techniques are incomplete, thus material that is revealed may be difficult to verify with any precision.

Event Analysis drives the Ptech method; events, processes, concepts, and, ultimately, classes are identified and defined. Moreover, this approach is consistent for all levels of abstraction. Ptech is a 'goal-directed, behaviour-driven' method integrating formally the structural and behavioural views of a realm of interest. An approach endorsed by Kent's view that OO analysis should focus on the operational requirements identified from behavioural analysis to determine the object structures and definitions (Kent, 1990).

The verdict on Ptech is also mixed. The focus on behaviour offers the domain expert an immediacy of interpretation that is denied with the more 'data-centric' methods. Also the Ptech approach remains consistent at all levels of abstraction thus relieving the domain expert of the need to introduce mediating cognitive strategies. That, at least, is how it should work in theory. In practice, the reality is somewhat different. Although the abstraction and codification conventions are conceptually sound, they are also extraordinarily complicated; originating, as they do, from the work of Russell and Whitehead. The cognitive investment for analyst and domain expert alike is thus considerable. The risk, therefore, with the Ptech method, is that any knowledge assets would remain fragmentary unless considerable effort is made to overcome the complexity.

#### 5.6.6 Object-Oriented Information Engineering (OOIE)

James Martin and James Odell first presented OOIE in 1992 (Martin & Odell, 1992). The presentation of the material is somewhat variable, but it appears the authors are attempting to forge a hybrid method from Ptech and IE. In addition to the core content of each method, object-flow diagrams are proffered as a technique to express the function or purpose of a process rather than the dynamics of its triggers, events and control conditions. They are designed to represent interactions between processes, without indicating when or how processes will be activated or terminated. 'Object-flow diagrams employ a strategic-level approach in the OO manner.' The following definition is provided: 'Each activity is a process that is performed to achieve a specific purpose. A product is the end result of that purpose. Activities consume saleable objects, add value to these objects, and produce a new product'.



Figure 5-8 Object flow diagram (Martin & Odell, 1992)

From these brief extracts, it is clear that an attempt is being made to incorporate the concept of Porter's value chain (Porter, 1985). The value chain is derived from representation of industries and enterprises as business systems and functions rather than individual activities (see also (Bower, 1973), (Gluck, 1980), (Buaron, 1981)). Enterprise value chains are derived from broader industry equivalents. Differences between competitor value chains provide a key source of competitive advantage (or threat) to which the enterprise must respond. Value chains comprise distinct value activities designed to create a product of value to its consumers, and a margin that is the difference between the product value and collective cost of performing the value activities. The generic value chain identifies value activities as either primary or support activities, where primary activities are concerned with the creation and delivery of the product to the consumer and support activities provide support to the primary activities and to each other. Porter argues that value activities form the building blocks of competitive advantage and the infrastructure of an enterprise should be designed to provide appropriate support for all value activities in its value chain.

Overall, the results of attempting to integrate Ptech, IE and Porter's value chain are unconvincing and the response has been generally unfavourable. Many of the presentation problems were resolved with a later publication (Martin & Odell, 1995). However, the content was purged of any reference to IE, and object-flow diagrams were relegated to a chapter addressing other modelling approaches. By 1996, object-flow diagrams have been restored to a position of relative prominence together with a new role (Martin & Odell, 1996). Object-flow diagrams are claimed to provide the means to transcend to the 'defined'-level of Humphrey's process maturity model (Humphrey, 1989). Template-driven design is identified as a technique employed by 'defined'-level organisations. Design templates also referred to as patterns providing a means of implementing an analysis model. The object-flow diagram is used to describe (and presumably manage) the process of template driven design. IE resurfaces again to present a contrast with the new concept of 'corporate object engineering' which appears to have a similar mandate to its predecessor but requires traditional developers to 'take off the blinders' and recognise there is 'more to the world than just data.' The top-down approach advocated by IE is not abandoned, but there is now a

concession that a bottom-up or middle-out approach may be preferable in some circumstances. Corporate object engineering provides for iterative development and concurrent engineering, while advocating 'reuse in the large', i.e. constructing business models and patterns with modular components. Finally, Martin and Odell claim that corporate object engineering represents an evolutionary transition from IE that will evolve itself to reflect future change.

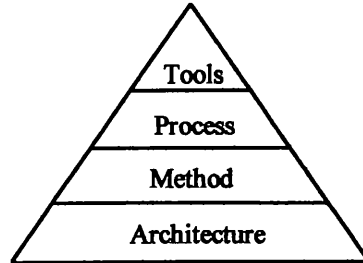
The focus of OOIE seems to be confused and diverse. While many of the concepts and techniques borrowed from other methods are sound, it is not at all clear how they blend together to form a coherent and cohesive method of analysis. It is not clear how, in this stage of its evolution, how OOIE can contribute meaningfully to organisational knowledge creation.

#### **5.6.7 The Object-Oriented Software Engineering (OOSE) method**

Jacobson (Jacobson et al, 1992) argues that software engineering must be considered an industrial process if software products are to be resilient to continuous and progressive change. Incremental development supported by prototyping is identified as the approach to managing the complexities of constantly changing requirements; and to achieve satisfactory levels of productivity, reusability must be factored in as an essential element of the development process. In Jacobson's view, while conventional techniques have failed to achieve reusability, OO provides new techniques with greater potential to support reuse. Effective support for continuous change can only be achieved if stable structures are developed to define an application domain, i.e. to achieve flexibility we must first achieve stability.

To define a system development process designed to support stability, Jacobson adapts the analogy of an industrial process to introduce the concept of an architecture defined as concepts, models and techniques forming an approach from a universe of approaches. Guidelines governing the application of these architectural elements to projects were developed as methods defining step-by-step procedures. Jacobson

recognises that methods are of little use when scaled up to support the entire life cycle of a product involving different types of project, and describes processes to manage the complete product life cycle. The concept of a software factory is viewed as a natural extension to the development process.

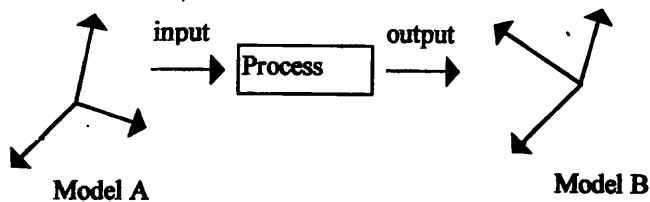


**Figure 5-9 The rational enterprise philosophy (Jacobson et al, 1992)**

Jacobson argues that OO provides the basic concepts to implement the rational enterprise philosophy; most significantly, environments and enterprises are viewed as collaborating objects exhibiting some prescribed behaviour. Therefore, OO provides the immediate benefit of potentially closing the semantic gap between perceived reality (i.e. the user model) and the system model.

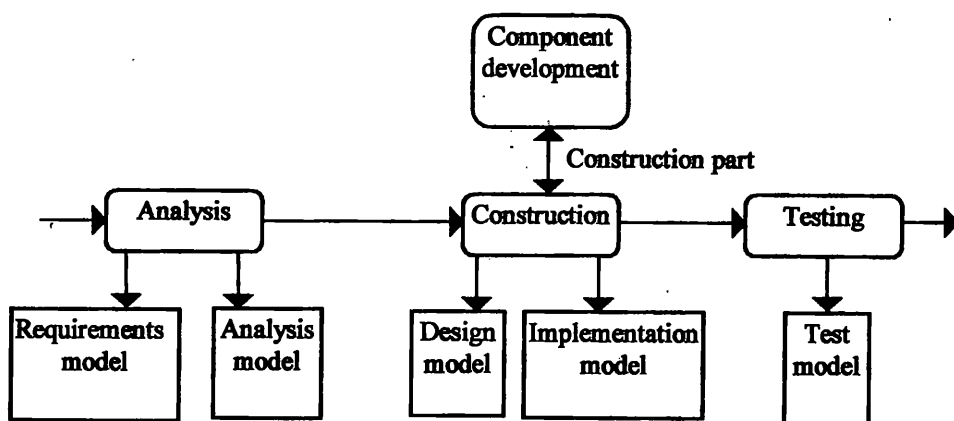
Jacobson utilises the concept of software entropy, which states that a program that is used will be modified and its complexity will increase (the combinatorial explosion of change) unless prevented, to illustrate the demands confronting software engineering methods. The conventional determinism (as defined by the waterfall model) to managing software entropy is considered to be inappropriate ('most of the water flows upwards'); Jacobson prefers the Boehm's spiral model (Boehm, 1986). Moreover, the prospects of major changes are reduced substantially as modifications are likely to be localised to stable and semantically intuitive objects.

For OOSE, system development is expressed as a process of developing a series of models to which complexity is added progressively throughout the development life cycle. A process is defined generically as the transformation of one model to another model:



**Figure 5-10 The process transformation (Jacobson et al, 1992)**

Jacobson seeks to provide a seamless interface between the models and to maintain traceability between the objects populating each model. The techniques defined to develop the model constitute the system architecture; where each technique is defined in terms of syntax, semantics and pragmatics. Methods are defined illustrating how to work with the techniques, while processes manage the model transformation between successive life cycle stages.



**Figure 5-11 The OOSE system development processes and associated models. (Jacobson et al, 1992)**

The analysis process is concerned with the production of a requirements model and an analysis model.



The requirements model is designed to reflect the user perspective by developing:

- a use case model
- interface descriptions
- a problem domain model.

The use case model (Jacobson *et al*, 1992; Jacobson, 1994a, 1994b, 1994c, 1995) is responsible for propelling Jacobson to a position of some prominence, but the technique has yet to achieve unanimity of purpose and interpretation (among practitioners at least). Indeed, conference presentations have been dedicated simply to achieving a consensus on the use and abuse of use cases (even the phrase 'use case' can prove problematic).

Quite simply a use case model consists of some system domain, the services provided by the system, and the users requesting the services. The services are represented by instances of use cases, while the users are represented as actors. Jacobson distinguishes between an actor that is viewed as a class and users that are instances of that class; a user can exist as an instance of multiple actors. Actors may also be divided into primary actors who use a system directly, and secondary actors providing support to primary actors interfacing with a system. A use case is defined further as a behaviourally related sequence of transactions forming a dialogue between an actor and a system. As a use case represents a complete flow in a system, it has a state and behaviour and therefore conforms to Jacobson's definition of an object. A use case is identified for every complete course of events initiated by an actor, and may be derived by extracting the following 'actor perspective' detail from the requirements specification:

- what are the principal tasks to be performed by each actor?
- what access to system information is required by each actor?
- is an actor required to inform the system of external changes?
- does an actor require details of unexpected changes?

The execution of each use case is then described in terms of the basic course detailing the principal activities, and multiple alternative courses defining any variants from the basic course. Jacobson provides an 'extend' association enabling the functionality of a use case to be extended by inserting another use case description, although the inclusion of this facility within an analysis domain is dubious.

Development of a user interface simulating use case functionality is viewed as essential in assisting the user to visualise the system. A system prototype must, of course, reflect the user's conceptual model to develop further the system concepts and semantics. Problem domain objects define the user's conceptual model. Jacobson claims that other OO methods concentrate exclusively on domain object models using heuristics to identify objects, while OOSE employs the use case driven approach as the core to developing all models.

The analysis model presents orthogonal views populated by the object types shown below; its purpose is to express the distribution of behaviour specified in the use case descriptions from the requirements model.

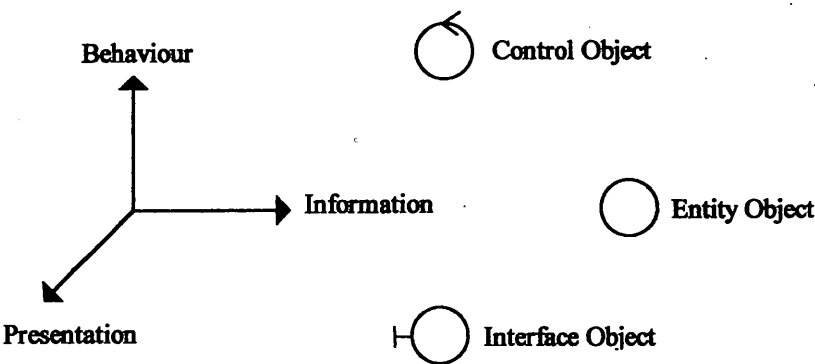


Figure 5-12 Jacobson's analysis model (Jacobson *et al*, 1992)

Entity objects represent persistent information; typically, information that persists beyond a use case invocation. While the problem domain object model may yield 'obvious' entity objects, Jacobson

recommends analysis of use case descriptions to identify all essential entity objects. The decision on whether to represent a piece of information as an entity object or attribute is based on how use cases manipulate the information. Where information is used separately it should be expressed as an entity object, and where information is strongly coupled with other information and never used in isolation it should be expressed as an attribute of an entity object. Jacobson prefers to defer the identification of entity object operations to the construction process when the structure stabilises, but stresses the necessity of distributing behaviour appropriately across interface objects, entity objects and control objects.

Control objects are reserved for complex use case behaviour that cannot be placed naturally in interface objects or entity objects; Jacobson notes that to somehow distribute complex behaviour over other object types (as suggested by other methods) compromises resilience to change. The primary roles for control objects are to manage event sequencing and communication between other object types. Preliminary control objects are assigned to each use case, to act as repositories for any behaviour not described by the interface and entity object types. The number of control objects for each use case is based on the number of actors interfacing with the use case (each actor should be assigned a control object) and a judgement on the level of complexity. While Jacobson's approach to determining the existence of control objects is somewhat heuristic, the principle is undoubtedly sound; when objects become overloaded with inappropriate behaviour, their encapsulation becomes compromised.

Interface objects express those requirements that depend directly on the target system environment, usually shown as a subset of the non-functional requirements. Jacobson identifies three sources for interface objects: system interface descriptions from the requirements model, analysis of the actors, and interface specific elements of the use case descriptions.

Jacobson suggests the preparation of a use case view narrative to ensure that all the object type roles and responsibilities collaborate to offer the use case functionality. The familiar aim of defining subsystems exhibiting strong cohesion and weak coupling is achieved by assigning subsystems to the principal control

objects and then clustering those interface and entity objects that have the greatest affinity (i.e. strongest association) with the control object.

For the purposes of this thesis, the construction and testing processes are of marginal interest. Nonetheless, Jacobson notes that construction and testing should be fully integrated with the analysis process.

How is OOSE to be appraised? To its credit, the 'use case' technique is a powerful and innovative device for capturing and expressing user requirements, which is being adopted increasingly in some form by other OO methods. Additionally, the associated concept of a 'control object' has contributed to an understanding of the structure of an object model. More contentiously, Jacobson has sought to transform software engineering into an industrial process providing methods to build large and complex systems, but has failed to demonstrate how OOSE is scaleable to the corporate level. Indeed, Jacobson reports 'Our experience of applying these techniques is very limited. This can be seen from the often schematic examples shown; the examples that we used when we clarified our ideas. We present our experience here to show that our work is capable of being scaled up, and to offer others the opportunity of extending it' (Jacobson *et al*, 1995). Again the familiar criticism that identification of objects depends simply on a grammatical analysis of use case narratives can be levelled at OOSE. Moreover, there is little guidance on defining object structures. Also, while Jacobson claims that OOSE reflects an industrial process in that it offers an orderly progression through the key processes, there is little reference to checks for consistency or completeness and the introduction of behavioural and non-functional requirements seems uncertain.

The use case and the three types of object are innovations that have the potential to contribute significantly to the knowledge creation process. They provide immediacy and descriptive capability to the domain expert. The abstraction and codification conventions for the use case, however, might prove problematic where an extended cognitive commitment is required from the domain expert. The concern is

that without some formalism to guide analysis, the contents of use cases might become somewhat arbitrary and ambiguous with an ensuing loss of utility.

### **5.6.8 The Booch Method**

Classification is a process of discovery and invention to identify the classes and objects belonging to an application domain, and therefore serves as the foundation of OO analysis and design according to Booch (1991; 1994). However, classification always involves consideration of contradictory and competing factors preventing description of a prescriptive approach to the process. Booch has thus adapted techniques and heuristics from disciplines such as biology and chemistry. In common with these disciplines, Shaw recognises that the construction of complex software systems also necessarily requires the classification process to be iterative and incremental (Shaw, 1989). Three approaches to classification provide a theoretical basis for OO analysis and design:

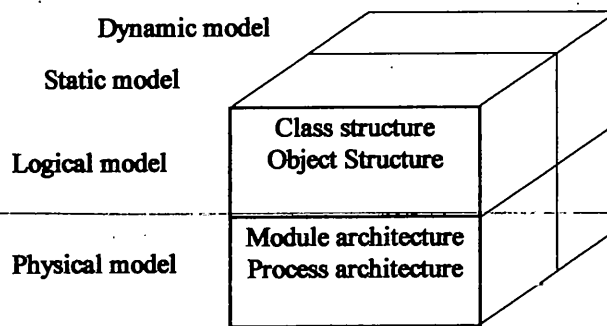
- classical categorisation
- conceptual clustering
- prototype theory.

Booch trawls the works of Plato, Aristotle, Aquinas, Descartes and Locke to determine the philosophical origins of classification theory and settles on a quotation from Aquinas that categorisation is a matter of clustering objects 'according to the knowledge we have of its nature from its properties and effects'. However, Kosko [Kosko, 1992] notes that as natural classifications tend to be 'fuzzy' and lack crisp boundaries, the classical approach to categorisation tends to be problematic. Conceptual clustering attempts to address these problems by probabilistically clustering objects according to conceptual descriptions (Stepp & Michalski, 1986). Objects may belong to multiple groups with varying degrees of fitness; conceptual clustering identifies the cluster with the 'best fit'. While classical categorisation and

conceptual clustering are viewed as adequate for the majority of classifications, cognitive psychology provides prototype theory (Lakoff, 1987) that endeavours to classify objects according to the extent to which they resemble a prototype.

Booch views analysis as a process of modeling the problem domain by discovering abstractions from the domain vocabulary while design invents abstractions and mechanisms to implement the required behaviour. From this foundation, Booch seeks to describe a method for conducting analysis and design; the method is divided broadly into a notation and a process, with a final section addressing issues.

Booch adopts the block metaphor shown below to describe the notation for the method:



**Figure 5-13 The Models of Object Oriented Development (Booch, 1991)**

With the block metaphor, Booch notes the necessity to develop logical, physical, static and dynamic semantics when analysing a complex system. Class, Object, Module and Process diagrams are developed to express the static perspective, while State Transition diagrams and Interaction diagrams represent the dynamic semantics.

**Class diagrams** represent a class structure view of a system in terms of class existence and their relationships in the logical design of the system. For the analysis phase, class diagrams represent class

roles and responsibilities supporting system behaviour, while the design phase concentrates on using class diagrams to capture the structures reflecting the system's architecture. Booch notes that each class should be supported by a non-graphical specification defining: responsibilities, attributes, operations and constraints ('a class or relationship invariant that must be preserved when the system is in a steady state').

Booch's variant on **state transition diagrams** is based on Harel's familiar notation (Harel, 1987), but with OO supplements from Rumbaugh *et al* (1991) and Bear *et al* (1990). A state transition diagram is developed during analysis to express the event-driven behaviour of an entire system, while design focuses on the dynamic behaviour of classes or class collaborations. State actions, conditional state transitions, nested states and state histories are available to enhance the expressive power of the diagrams.

**Object diagrams** represent scenarios by expressing object existence and their relationships in the logical design of the system; they are a prototypical representation of interactions and structural relationships that exist independently of any specific object collaboration. Key elements of an object diagram are restructured to express the sequencing and synchronisation of a scenario.

---

**Module diagrams** are used optionally in physical design to indicate the physical layering and partitioning of a system architecture by representing the allocation of classes and objects to modules. Modules and their dependencies populate module diagrams. Where the number of modules indicates that additional partitioning may be appropriate, modules may be grouped into logically cohesive clusters and viewed as subsystems. **Process diagrams** are used in physical design to represent the allocation of processes to the physical collection of processors, devices and connections that serve as the technical platform, where a processor is defined as a 'piece of hardware capable of executing programs'.

The Booch process is based neither on an 'anarchic' nor deterministic approach to building system architectures. In preference to 'top-down' or 'bottom-up' system development, Booch opts for an iterative and incremental development life cycle based on the concept of 'round-trip gestalt design'. Booch argues

that a well-managed iterative and incremental development life cycle must balance the requirement to achieve process maturity (Humphrey, 1989) with the need to stimulate creativity and innovation by 'faking' a rational process (Parnas & Clements, 1986).

Booch seeks to resolve this apparent paradox by introducing the concept of micro and macro development processes which marks a significant advance on previous attempts to describe a comprehensive development process (Booch, 1991). The micro development process is based on Boehm's spiral model of development (Boehm, 1986) and addresses the iterative and incremental approach to developing architectural products based on scenario definitions. It is designed to create the conditions necessary for 'opportunistic control' (see also (Stroustrup, 1991)). The macro development process is based on the traditional waterfall life cycle and provides the framework for controlling the micro process and managing risk.

The micro development process relies heavily on brain-storming, storyboarding and CRC cards (Beck and Cunningham, 1989) to identify classes, objects and their semantics. Behaviour analysis, based on system function points (an end-user business function) is also available as a technique (Dreger, 1989; Rubin and Goldberg, 1992). Finally, Jacobson's use case technique is selected to conduct scenario analysis for each system function point. As the semantics are explored in more detail, Booch exhorts the analyst to conduct 'pattern scavenging' as an approach to identifying commonality and revealing opportunities for reuse.

The macro development process consists of the following activity:

- establish the core requirements for the software
- develop a model of the system's behaviour
- create an architecture for the implementation
- evolve the implementation through successive refinement
- manage post-delivery evolution.



Establishing core requirements is interpreted as the conceptualisation process. This involves establishing a set of goals for the 'proof of concept' and evaluating the resultant prototype to determine whether the project should proceed. Booch adopts the view that describing the function of a system defines the objective of an analysis exercise (DeChampeaux *et al*, 1992). CRC cards and scenario storyboarding are employed to capture the semantics of the system. For the purposes of the thesis, the subsequent phases are of marginal interest.

Booch provides many interesting insights into OO, and yet the method fails as a credible OO analysis method. Indeed, the claim that analysis has been incorporated successfully into the method must be challenged. The most serious deficiency is the lack of a process to guide the method. While the concept of a macro-process and micro-process appears to be a plausible approach to achieving the advantages of providing a conceptual framework and facilitating 'opportunistic control', the reality is unconvincing. The whole edifice is based on the system function point and its associated scenarios; however, precisely how system function points are identified and defined remains unclear. Unless system function points are supported by a commonly accepted and verifiable definition (an 'end-user business function' is simply inadequate), the interaction between macro-processes and micro-processes remains uncertain. The phrase 'round-trip gestalt design' suggests pretensions to theoretical credibility that are probably unsubstantiated. A further weakness with the Booch method is the disproportionate reliance on heuristics to identify system structure and behaviour. It is only when basic structures and behaviour have been established that more formal techniques are introduced to consolidate and rectify heuristic content. The problem with heuristics is that anomalies may remain undetected until an advanced stage of analysis, thus necessitating considerably more rework than if formal techniques had been adopted earlier. At some threshold, reliance on heuristics therefore becomes counter-productive; the Booch method is at risk of exceeding that threshold.

The Booch method may therefore be effective to launch the knowledge creation process but it is unlikely that the resulting knowledge assets enjoy sufficient commitment to support subsequent cycles of externalization and internalization.

#### **5.6.9 The Unified Modelling Language (UML)**

UML represents an attempt to unify the methods of Booch, Rumbaugh and Jacobson; in addition, CRCs (Beck and Cunningham, (1989)) and Wirfs-Brock's stereotypes (Wirfs-Brock *et al*, 1990) are included as complementary options. A distinction is made between a modelling language and a process. A modelling language provides the focus of UML that is constituted of a notation and meta-model that defines the notation. Jacobson is revising the Objectory framework to define the unified process and has provided a brief account as an interim measure (Jacobson, Griss, Jonsson, 1997).

Booch, Rumbaugh and Jacobson are to produce a series of reference books describing UML (Booch, Rumbaugh and Jacobson, 1998; Rumbaugh, Booch and Jacobson, 1998; Jacobson, Booch and Rumbaugh, 1998). Fowler and Scott have produced an introduction to UML (Fowler and Scott, 1997) with Eriksson and Penker providing a more detailed account (Eriksson & Penker, 1998).

Objectory defines a development lifecycle comprising four phases:

- An **Inception** phase, where a business rationale and sponsorship is established for the project. For low-ceremony projects, inception requires little more than an informal agreement. While inception may demand a fully-fledged feasibility study for a high-ceremony project.
- An **Elaboration** phase, where detailed requirements are captured, high-level analysis and design is conducted to establish baseline architectures, and plans for the construction phase are created. Domain models and use cases capture functional requirements.

- A **Construction** phase where incremental iterations deliver solutions to project requirements. Each iteration includes the familiar activities of analysis, design, implementation and testing. Analysis models explore the implications of the functional requirements for particular applications.
- The **Transition** phase includes beta-testing, performance tuning and user training for deliverables produced from iterations within the Construction phase.

Jacobson's 'use case' diagram has assumed a central role in UML and forms the core of the Elaboration phase. Use cases are employed to capture requirements, and plan and manage project development. Fowler identifies the key distinction between a user goal and a system interaction when applying use cases to a domain. From the analysis perspective, interest is centred on the user goal as this provides the motivation for the actor invoking the use case. In all other essential respects, the concept of the 'use case' as a scenario modelling technique has remained unchanged for UML.

Following Cook and Daniels, UML declares that analysis is responsible for constructing the conceptual class model (Cook and Daniels, 1994). Familiar definitions are provided for classes, associations, attributes, operations, methods and generalisation. Attempts are being made to incorporate Wirfs-Brocks stereotypes into Jacobson's concept of control objects to distribute behaviour more effectively. However, as Fowler concedes, there is some confusion as to what actually constitutes a stereotype. Full expression is given to the classification options (for the conceptual level at least); single, multiple, static and dynamic classifications are all permitted. A fine distinction is drawn between aggregation and composition. With composition, the part object may belong to only one whole with parts usually expected to 'live and die' with the whole; this restriction, presumably, does not apply to aggregations. Furthermore, provision is made for reference objects, value objects, multi-value roles and immutability of attributes, roles and classes.

The notation adopted for Interaction Diagrams is the sequence diagram, derived largely from Buschmann's POSA diagrams (Buschmann *et al*, 1996), and the collaboration diagram. While the

sequence diagram is designed to show a sequence of events, the collaboration diagram illustrates how objects are connected statically.

Packages are the grouping mechanism adopted by UML to divide large systems into cohesive subdivisions; the concept is derived directly from Booch's categories (Booch, 1994). Dependencies between packages provide the heuristic for identifying and defining packages and their constituent classes.

UML utilises the familiar State Diagram conventions based on Harel's statecharts (Harel, 1987).

As Fowler notes, unlike the other techniques, the Activity Diagrams have not been employed in the previous work of Booch, Rumbaugh or Jacobson. It is claimed that Activity Diagrams are derived from event diagrams, state modelling techniques and Petri nets. The diagrams are considered useful when conducting workflow analysis and describing parallel processes. The diagramming conventions provide for parallel processing to commence and terminate with synchronisation bars. Crucially, activity diagrams can be used to describe complicated use cases. The facility is crucial, because the degrees of freedom conferred on an activity diagram ('An activity diagram need not have a defined end point.') impose no constraint on the existence of a use case; a situation that could constitute a potent source of ambiguity and confusion. Moreover, it is claimed that an activity diagram may span multiple use cases; the relationship between an activity diagram and a use case thus becomes more uncertain. Another innovation is the introduction of 'swimlanes' which provide for an activity diagram to be (re)arranged within swimlanes such that responsibility for a set of activities can be assigned to an organisation unit by extending the diagrammatic conventions. However, as Fowler concedes, they can become confusing on complex diagrams. Fowler identifies the following situations where activity diagrams may make a contribution:

- analysing the actions and behavioural dependencies required for a use case
- understanding workflow across multiple use cases, i.e. where use cases interact with each other
- dealing with multi-threaded applications.

The remaining material addresses design and programming issues, and is therefore beyond the scope of the thesis.

## 5.7 OO methods - an evaluation

The list of OO methods selected for evaluation is far from exhaustive; there are now some 70 published OO methods, other prominent methods which merit a brief mention are:

- **Shlaer/Mellor:** The Shlaer/Mellor method (Shlaer & Mellor, 1988; 1992) remains problematic as an OO method. Originally, they presented object models as little more than object extensions to entity models; references to encapsulation and inheritance were completely absent from the description. This oversight was later rectified, but other problems persisted with their relational approach to describing objects, e.g. the relational model is purely syntactic and fails to capture the semantics of a domain. The method is based around the three familiar, orthogonal dimensions: an object model; a state model; and data flow diagrams to describe the process model. There is still some debate about whether the Shlaer/Mellor method fulfills its claim to be object oriented.
- **OORAM:** Object-Oriented Role Analysis and Modelling (Reenskaug *et al*, 1996) utilises the role model as its basic object abstraction. From the OORAM perspective, real world phenomena are described by some constellation of collaborating objects. An object describing the domain of a phenomenon can play several roles. This fact is exploited by partitioning a domain systematically into separate areas of concern where a different role model describes each partition. Conversely, a series of role models may be synthesised to describe a complex domain. Reenskaug claims that role models invite separation of concern and thus facilitate a 'divide and conquer' approach to constructing domain models. The risk of fragmentation (where each role model describes only a limited aspect of the

domain) for large domains is avoided by role model synthesis, where derived role models for the complete domain are constructed from multiple base role models.

- **KISS:** The KISS method was developed by Kristen to adapt the basic principles of OO to capture information and meaning from everyday language (Kristen, 1994). Kristen argues that a conceptual model for a domain can be developed completely from a semantic analysis of the testimony of domain experts using an 'out-of-the-middle' approach. An object is perceived to have three dimensions: 'action', 'attribute' and 'time', and from this basis it is claimed that provision is made for the management of information. Furthermore an OO information system is constructed in three layers: the outer 'function' layer containing input and output functions; the middle 'structure' layer containing actions and objects, and the inner 'core' layer containing attributes and operations. Much is made of the necessity to eliminate any technical content from the definition of technical architectures; this, of course, has been a central tenet of analysis for several years. The emphasis on information systems militates against KISS resonating as a fully credible OO method.
- 
- **Syntropy:** Syntropy is presented as a 'second generation' OO method (Cook and Daniels, 1994). The exploitation of models is paramount to all aspects of Syntropy. It is recognised that although models often yield much contextual information, they also have implicit simplifying assumptions and thus merely provide a 'map' with which to navigate a domain. The concept domain (the domain of concern) describes the concept model for the situation in the real world that is the subject of the analysis. The concept domain is surrounded by interaction domains, i.e. an object management domain, a user-interface domain and a communication domain, which have responsibility for ensuring alignment between the concept domain and the real-world. The essential model is provided to support the analysis of the concept domain; an essential model is defined as a description of some real-world situation which is constituted of object configurations, events which cause state changes and the possible consequences of those events. The facts about a situation which are described by an essential model are as follows:

- the possible states in which a situation can exist (expressed as objects, properties, and relationships)
- the set of events which can cause changes between one state and another
- the possible sequences of events that may occur.

Syntropy clearly falls into the 'process-driven' stream of OO analysis methods, with much emphasis placed on event analysis. However, the definition of an event appears to be fundamentally problematic: i.e. 'several objects may change their state in response to a single event'. By failing to insist that an event may only change the state of a single object, the concept of an event is critically flawed with the result that multiple events may remain concealed and thus compromise the analysis effort.

- **The OPEN Method:** The OPEN (Object-oriented Process, Environment and Notation) Method is an international standardised OO development method (Graham, Henderson-Sellers & Younessi, 1997).

An international consortium in response to a call from the Object Management Group (OMG) developed it for a standardised OO metamodel and notation. The OPEN Method consortium appears to have provided a viable alternative to UML.

Each Software Engineering Process (SEP) is described by 'contract driven' based Activity objects, where each contract is defined in terms of pre- and post-conditions and invariants. Henderson-Sellers claims that the Open Process permits ordering in an iterative and incremental fashion, and tailoring to individual standards and culture (Henderson-Sellers, 1997). Moreover, the OPEN Architecture is claimed to illustrate an instantiation of an Architecture Reference Model (Hertha, 1995) designed to complement the OMG 'application' level architectures with a business architecture. The Architecture Reference Model comprises four layers:

- a Business Strategy Layer (providing declarations of strategic objectives and goals)

- a Business Process Layer (implementing the Business Strategy in the form of work flows)
- a Business Components Layer (automating part or all of a Business Process)
- a Technical Infrastructure Layer (the technical platform upon which business components execute).

The Architecture Reference Model is a potentially interesting (although somewhat skeletal) attempt to confer some scalability onto OO analysis methods. Similarly it will be interesting to monitor how scalability is factored into future evolutions of the OPEN Method.

Many would doubtless contend that OO marks a major advance over Structured Analysis and IE; and yet there is little evidence of significant or sustained productivity gains. Unquestionably, inept management, poor planning and lack of training still contrive to subvert system development projects; however, the anxiety that OO methods are deficient as viable approaches to analysis cannot be dismissed.

The range of OO methods examined in the thesis vary from those consisting of little more than a loosely coupled set of heuristics with little OO content, to more formal methods deeply immersed in the theoretical foundations of OO. It might appear from perfunctory consideration that either extreme of approach might be preferable in certain circumstances.

The growing complexity and turbulence of system domains is transforming the process of identifying and articulating requirements into an increasingly uncertain activity. To simply rely on some heuristic based on grammatical parsing of a requirements statement from which to define the structure and behaviour of a domain poses the risk of creating anomalous knowledge assets. Moreover, it is not clear how this approach is scalable to the corporate dimension. The approach appears only to be viable where a domain is simple and static, and well understood by the user community. Far more common is the situation where a domain is complex and turbulent, and the user community is highly uncertain of their requirements. OO methods offer the capability of confronting these circumstances with some degree of confidence, but it is necessary to constrain the use of heuristic methods and turn to the more formal techniques based on some



form of system development lifecycle. Of the methods selected for consideration for the thesis, there are four possible candidates:

- Edward's Ptech
- Jacobson's OOSE
- Booch, Rumbaugh and Jacobson's UML
- The OPEN Method.

With Ptech, there is great emphasis on the behaviour of a domain yielding the structure, with event analysis disclosing the concepts and classes describing the domain structure. With this approach, it is no longer necessary to simply infer structure from some semantic rule of thumb; the structure can be confirmed verifiably in terms of its support for the domain behaviour. Moreover, considering domain behaviour presents far less of a cognitive barrier for the end-user, the behaviour of a system is nothing less than the interface the system presents to its environment; the structure describes how that system is implemented. The behaviour-driven approach is thus completely analogous with the OO paradigm. Jacobson's use case is consistent with this approach although, it will be argued in the next chapter, the precise role of the use case requires some revision. The Unified Modeling Process (soon to be published as the Rational Unified Process (Jacobson *et al*, 1999)) is selected because the use case is included in UML, although the approach to activity modelling gives some cause for concern. Finally, the OPEN Method is selected because an attempt has been made to address scalability, and business components are subjugated to business processes.

From the preceding synopsis, it is clear that OO methods remain poised to make a major contribution to system development. The emergence of domain behaviour as the primary focus for eliciting requirements improves the prospects for establishing OO as a viable approach to analysis.

Much ground has been covered in exploring how the selected analysis might contribute to the knowledge creation cycle. It would be inappropriate to dismiss the non-OO methods simply because they now appear primitive in comparison with OO methods. It should be remembered that non-OO methods had much to contribute when organisations were permitted to pursue a more orderly existence. With the growing intensity of competition, however, that orderly existence has been replaced by one of chaos. Organisations have sought to counter this destabilizing influence by following a trajectory of complexity between the influences of order and chaos. It is in this mode that non-OO methods have been less effective and OO methods were poised to make a greater contribution. It is possible to conclude therefore that non-OO methods have much to contribute to the aspects of an organisation that are unitary and mechanistic, while OO methods are better placed to tackle the greater complexity and uncertainty that comes with the more pluralistic and organic areas. It is in these latter areas that organisations exist in a state of complexity and must endlessly adapt if they are not to fall victim to the competing influences of order or chaos. Complex systems, however, display emergent patterns of creative adaptive behaviour and it is this feature that may allow OO methods to eventually fulfill its potential. It should also be noted that the emergence of these patterns is predictable in a complex domain. The ability to predict is of course dependent on the available knowledge about a domain.

One area where knowledge is sought is the flux of system taxonomies that define the dynamics of a domain. Here OO has greatly increased the variety of system taxonomies that can be analysed using some form of standard technique. Techniques are now available to analyse systems that may be viewed as simple configurations of collaborating objects, systems with behaviour that emerges though different levels of hierarchy, and systems that exhibit recursive structures.

In theory, the richness of expression of the abstraction and codification conventions should allow complex concepts to be expressed with precision and clarity. Furthermore, the tools conferred on the analyst and domain expert should provide for fertile iterations of externalization and internalisation, with complex webs of collaborating objects providing some protection against erosion of value though their ability to

reform to exhibit new behaviour. And yet a feeling of unrealized potential pervades the OO industry; the possible reasons for this are discussed in the next section.

## **5.8 Analysis methods - a final evaluation**

The earlier exploration into the origin and nature of requirements concluded that they emerge from knowledge assets when an organisation is in a state of complexity. Knowledge assets are formed tentatively when attempting to adapt to the competing forces of order and chaos. The struggle between these forces creates shifts in the patterns of behaviour of the organisation that are difficult to detect but are, nonetheless, predictable. It is not surprising, therefore, that the resulting knowledge assets may be elusive and difficult to articulate. Indeed they may be held in a tacit state and remain so until articulation is possible.

A contribution of this thesis is to contend that analysis should focus attention on these tacit knowledge assets, as it is here where competitive advantage is likely to be gained or lost. It has also been argued in the thesis that knowledge creation is not a discrete activity but must be viewed as an endless commitment to the externalization and internalisation of domain expert knowledge. This commitment extends further the focus of analysis. It is not clear that the analysis methods reviewed in this chapter can achieve this focus. The reasoning behind this claim is detailed in the following paragraphs.

It has been stated previously that the expert first apprehends and then interprets a change in a domain by shifts in the patterns of behaviour of the domain or its external environment. Behaviour is thus a key cognitive concept around which much reasoning is based. And yet it is with this very concept that many analysis methods introduce ambiguity. To fail to devise robust abstraction and codification conventions to express behaviour is to present the domain expert with significant cognitive barriers at some point in the analysis process. A definition of organisational behaviour is provided in Chapter 6.

Domains of complexity exhibit patterns of behaviour that shift in ways that are difficult to predict. However, it is through the interpretation of knowledge assets that the domain expert is able to observe 'weak' signals and interpret their significance in predicting future shifts in patterns of behaviour. It is through this process that new knowledge assets are created and requirements for change evolve. The implication here is that behaviour is not observed as single nodes of activity, but rather as shifting patterns of activity formed from multiple nodes. To ignore the way the domain experts exploit these patterns is to neglect a fundamental cognitive strategy and risk overlooking knowledge asset configurations essential to the domain under scrutiny. The externalization and internalisation processes are therefore placed under threat. To return to Boisot for a moment, the SLC is likely to become distorted or aborted. Any analysis method seeking to contribute to the knowledge creation must embrace the detection and interpretation of patterns, and in particular patterns of behaviour. This discussed in Chapter 7.

## **Chapter 6**

### **6 The Role of the Business Process**

#### **6.1 Introduction**

Knowledge assets are the source of requirements with the greatest potential to confer competitive vigour on an organisation. Moreover, knowledge assets are formed when domain experts respond to 'weak' signals that may presage a shift in the patterns of behaviour exhibited by the organisation or its external environment. It is curious therefore that this aspect of analysis remains resolutely the area of least efficacy for the majority of mainstream methods.

The review of analysis methods presented in the previous chapter offers the conclusion that none are capable of implementing the mandate to propagate the knowledge creation cycle. From the sample of analysis methods subject to review against this criterion, it is argued that the methods listed below had the potential to contribute to the fulfillment of the mandate:

- Ptech
- OOSE
- UML (and the Rational Unified Process)
- The OPEN Method.

The methods are selected on the basis of their OO foundations and focus on behaviour (as opposed to structure) as the primary focus of analysis. It is argued that shifts in the interrelationship between an organisation and its environment are manifested in their respective forms of behaviour (structure serves merely to implement behaviour). It is also claimed that enduring inconsistency in adequately defining those activities constituting organisational behaviour has ensured that analysis remains problematic.

From these arguments, it may be concluded that a unit of activity must be devised that allows the analyst to penetrate tacit knowledge. The argument is presented in this chapter that the 'business process' is the desired unit of activity. However, the selection of the business process presents a variety of semantic and conceptual difficulties. Not least amongst these has been the recent proliferation of management theories on process redesign, which may be referred to collectively as Business Process Reengineering (BPR). Each theory presents an apparently plausible definition of a business process; but, frequently, these definitions prove to be inadequate and ambiguous in practice. This chapter examines how Porter's value chain (Porter, 1985), may be adapted to provide a definition of a business process that is semantically and conceptually consistent with the reality of organisational behaviour. Discovering and defining processes presents another difficulty for analysing organisational behaviour. It is not sufficient to observe activities being performed in a domain and simply to deduce the existence of business processes from these activities; this approach resonates with the worst excesses of the heuristic approaches to analysis. What is needed is an approach that verifiably discovers and defines a business process, and reflects the interdependencies between processes. A technique for analysing processes such that these requirements are satisfied is presented in this chapter. Finally, the way in which business processes are frequently implemented in an organisation ensures that essential details remain concealed or are distorted in some fashion. By embracing the concept of 'separation of concern', it is demonstrated how the subverting influence of organisations may be eliminated from discovering and defining business processes.

The conclusion is that the study of domain behaviour provides the most direct and unmediated route to tacit knowledge. However, this alone is insufficient. To detect and evaluate subtle shifts or variations in domain behaviour, it is necessary to discover essential patterns of behaviour. The generation of tacit knowledge should be focused on shifts in the patterns of behaviour, other shifts may be incidental and not require a strategic response from the business. These considerations form the basis of the chapter on analysis patterns.

## **6.2 What is a Business Process?**

A process represents a single example of a domain activity. Business systems and functions are also examples of domain activities. Before attempting to define a business process, it is thus helpful to distinguish it from business systems and functions. Business systems are based on the concept of an organisation arranged as a series of collaborating business systems, each comprising a hierarchical arrangement of functions (Bower (1973), Gluck (1980), and Bauron (1981)). The idea is that each business system is formed from some cohesive cluster of functions based around a thematic core, e.g. Sales, Distribution, Research and Development, Accounting, Finance. From this definition of a business system, it is possible to proceed further and venture a definition of a function. Martin provides the following definition (Martin, 1986):

- An enterprise function is a group of activities which together support one aspect of furthering the mission of the enterprise.
- A function is ongoing and continuous.
- A function is not based on organisational structures.
- A function categorises what is done, and not how.

Furthermore, Martin provides the following definition of a business process:

- A process is a specified activity that is repeatedly executed in an enterprise
- A process can be described in terms of inputs and outputs.
- A process has a definable start and stop.
- A process is not based on organisational structures.
- A process identifies what is done, and not how.

These definitions raise some interesting points and, in fact, are superior to many later definitions. The most obvious point is that a function (and by implication, a process) must further the mission of the enterprise; this is a slightly different focus from the BPR emphasis on 'customer-centric' considerations. Whereas a function is defined to be ongoing and continuous, a process is a tangible activity with clearly defined entry and exit conditions, and is executed over a finite period. Also, Martin notes two features that are shared by functions and processes: both are independent of organisational structures and both describe logical activities. By insisting that functions and processes are independent of organisational units, Martin is seeking to avoid distortion of their respective definitions; in fact, those organisational units associated with some aspect of a function or process are recorded separately on a distribution matrix. Martin extends further the concept of function and process independence by insisting that their definitions are free from any considerations of implementation. These are included later when procedures are designed to implement the functions and processes. However, as discussed in the previous chapter, Martin's definition is undermined by the subordination of process to function.

Porter adopts a somewhat different approach to defining business activities by using the value chain to define primary and support activities (Porter, 1985). Primary activities are defined as 'the activities involved in the physical creation of the product and its sale and transfer to the buyer as well as after-sales assistance'. Porter identifies five generic categories of primary activity: inbound logistics, operations, outbound logistics, sales and marketing, and service (to enhance or maintain the value of the product). Support activities are defined as those activities that 'support the primary activities and each other by providing purchased inputs, technology, human resources, and various firmwide functions. Porter provides four categories of support activity: procurement, technology development, human resource management, and firm infrastructure. Moreover, within each category of primary and support activity, Porter reports that there are three activity types that contribute differently to achieving competitive advantage: direct activities that create value for the customer; indirect activities that make it possible to perform direct activities on a continuing basis; and quality assurance. Although, Porter does not refer



explicitly to a function or process, the primary and support activities reflect the concept of a business process.

These definitions precede the advent of the BPR 'revolution' which sought to reinvent an organisation to varying extents around the concept of the business process. Early attempts to define a business process are listed below:

- The logical organisation of people, materials, energy, equipment, and procedures into work activities designed to produce a specified end result. (Pall, 1987)
- A set of logically related tasks performed to achieve a defined business outcome. (Davenport and Short, 1990)
- A process is simply a structured, measured set of activities designed to produce a specified output for a particular customer or market. It implies a strong emphasis on how work is done in an organisation, in contrast to a product's emphasis on what is done. A process is thus a specific ordering of work activities across time and place, with a beginning, an end, and clearly identified inputs and outputs: a structure for action. (Davenport, 1993)
- A collection of activities that takes one or more kinds of input and creates an output that is of value to the customer. (Hammer and Champy, 1993)
- A business process is identified if:
  - it has specific inputs and outputs
  - it crosses organisational boundaries
  - it focuses on goals and ends

- it has comprehensible inputs, outputs and description
- it relates to customers and their needs. (Hammer and Stanton, 1995)
- A set of linked activities that take an input and transform it to create an output. Ideally, the transformation that occurs in the process should add value to the input and create an output that is more useful and effective to the recipient either upstream or downstream. (Johansson *et al*, 1993)
- Ould defines a process is to be a purposeful activity, it is done collaboratively, by a group, often having cross-functional boundaries, and is invariably driven by customers. Processes are of three types: core processes which concentrate on satisfying customer needs, support processes concentrating on satisfying internal customer needs, and management processes concerned with managing the core and support processes. (Ould, 1995)

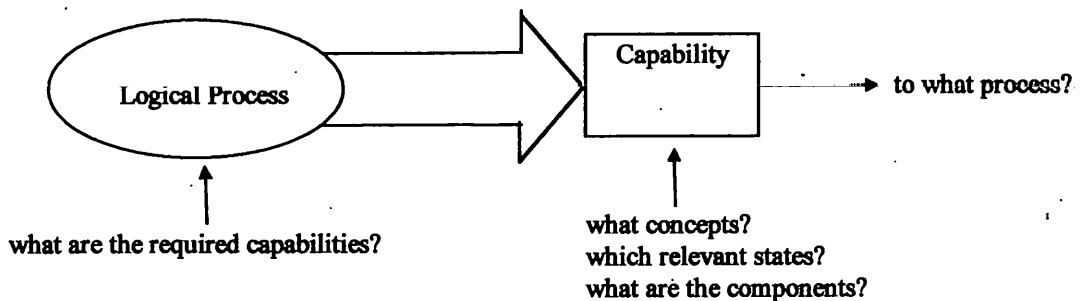
For Darnton these definitions are not particularly deficient but it is noted that they do not draw from other cultures such as management cybernetics (Beer (1985); Espejo and Harnden (1989)). However, although these definitions appear to embrace the core themes of analysing business processes, there is no single definition that is sufficient to support the identification and definition of business processes. For a more rigorous definition of a business process that may actually sustain detailed analysis, it is necessary to incorporate the issues raised in the definition provide by Halé (Halé, 1995).

Halé has devised a Value Analysis technique that is claimed to produce a logical model of an enterprise or system that is non-redundant: 'a process will appear only once in the model with all its relevant concepts, business objects, events and roles'. Moreover, 'the model is the basis for the definition of the activities, roles and logistical resources which can be implemented in the reality of the business.' Halé contends that Value Analysis produces processes that are 'reusable across the enterprise as a whole'.

### 6.3 The Value Analysis approach to defining processes

Halé's Value Model draws heavily on Porter's Value Chain in that they both focus on the value of the end-product provided by a process. However, there is a crucial distinction. Whereas Porter's value chain is concerned with monitoring the efficiency of a process, Halé analyses the effectiveness of a process. To monitor the effectiveness of a process, Halé defines value to be an element of the product delivered by the process which expresses a capability provided to its user. The Value Analysis technique consists of the following steps:

- understanding the purpose of the end-product
- deriving the capability that it provides
- understanding the components of that capability - the concepts which are necessary and sufficient for defining these capabilities
- identifying the states of the components which are relevant to the process.



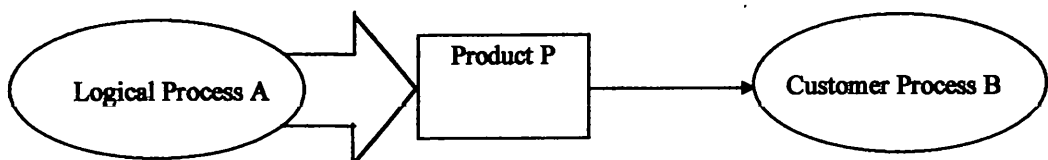
**Figure 6-1 The End-product as a Capability (from Halé)**

Halé acknowledges the influence of the Ptech method and claims that the Value Analysis technique has integrated the essence of Ptech into a complete BPR approach.

The Value Analysis notation depicts processes adding value to one or more products to deliver a single end-product; a product is synonymous with a capability defined earlier. At any level of abstraction, a process will only produce a single end-product which represents the value added by the process. Each process has only one end-product and a given end-product is delivered by only one process. The definition of a process is constrained further by insisting that the end-product will always benefit the customer of the process (even when the customer is another process); processes are chained by allowing the end-product of one process to be the resource for another process. As Halé notes, the end-product defines completely the interface between two logical processes:

Halé suggests a possible protocol of communication between the two processes shown in the following diagram:

- process B requests from process A a product P
- ~~processes A and B agree the particular shape, status, delay, and cost, i.e. the state of the required~~  
product
- process A commits to delivery
- process B may supply money or other resources to A for completing the transaction
- the exchange takes place
- all consequences and loose ends are resolved.



**Figure 6-2 A protocol of communication between two processes.**

A corollary of the preceding definitions is that a product defines the complete set of transactions between two processes. Halé exhorts the analyst not to confuse the Value-Added Diagram with any form of 'flow' diagram; there is nothing flowing between the processes. The arrows in the diagram represent dependencies between the processes, and associations between processes and products. A process specifies the capabilities it needs and the capability it can provide. Although a process is constrained to deliver only one end-product, it may need some combination of products from other processes before it can execute, i.e. a product may be used by several processes.

A process, its end-product and its resource-products (products required by the process) are together deemed to constitute the domain of a process. Halé asserts that this separation of concern is a powerful technique of simplification where many of the temporal and synchronisation requirements remain hidden in the product structures and definitions, i.e. in the business objects and their associated states. The Value-Added diagram also allows for an abstract process to be decomposed into processes of finer granularity providing the end-product is also the end-product of one of the sub-processes and that each of the resource products is used by one or more of the sub-processes. As Halé notes, the key to analysing processes is to enforce the formalism that each process has a discrete domain and no knowledge of any other process. This concept is, of course, completely analogous with the basic tenets of OO.

## **6.4 The Analysis of Business Processes**

Approaches to analyzing business processes exhibit considerable variety. On one hand, definitions are bereft of any formalism and any activity may be interpreted as a process. There is no attempt to distinguish between logical and physical activities, or to extricate the definition of an activity from the context of the organisational framework in which it is executed. Moreover, the rules for the inclusion, exclusion and association of business activities are not specified. These definitions offer such little

guidance on identifying and defining business processes that they merit no further consideration. On the other hand, there are definitions infused with significant formalism and based on sound methodological foundations. Selection of the formal approach to defining business processes is based on the following rationale:

- The notation and underlying concepts do not require a major cognitive investment. The emphasis on 'separation of concern' provides an approach to managing complexity; paradoxically, the more simple definitions provide no such mechanism and the ensuing complexity can pose cognitive barriers. Moreover, the consistent delivery of verifiable business processes serves to reinforce the underlying concepts and thus reward cognitive investment.
- Process definitions are characterised by precision and clarity. Furthermore, the definitions yield an abundance of supplementary information: the associated business events, the associated concepts and business objects; and myriad types of association with organisational units, strategic declarations, and technical constraints

If business processes are to be analysed, the arguments for adopting a formal approach appear to be highly persuasive, if not conclusive. While it is not argued that analysing the structure of business processes will capture all facets of an organisation, a comprehensive process model for an organisation provides a robust basis for exploring other, 'softer' issues. In particular, the socio-technical concerns of Mumford's Ethical method (Mumford, 1996) may be articulated more precisely if 'human-centric' requirements can be assigned to individual processes. Similarly, the messy, ill-structured problems that provide the focus for Checkland's SSM (Checkland, 1981) may also benefit from an unambiguous declaration of business processes. It remains to review approaches to analysing business processes based on the formal expression of their definition.

## **6.5 The Identification and Definition of Business Processes**

A contribution of the thesis is to propose a new approach to the analysis of business processes that also offers the possibility of contributing to the knowledge creation cycle. The motivation for this initiative is that a failure to fully articulate business processes may present significant barriers to the sustainable elicitation of tacit knowledge.

To understand fully the essence of a business process, it is necessary to explore organisational behaviour. At an abstract level, the behaviour of an organisation represents an interaction with its environment. Contrary to the approaches advocated by the more data-centric methods, changes in environmental or organisational influences are detected by shifts in behaviour and not structure; structure is merely some artifact to implement behaviour. The creation of tacit knowledge is an initial response to the detection and contemplation of a shift in the behaviour of an organisation or its environment.

---

From the work of Halé, it can be inferred that the behaviour of a domain may be expressed as a web of events and responses to those events. In fact, it is stated that events capture the dynamics of the business objects populating the domain. The dynamics of a business object are expressed as:

- the creation of an instance
- the termination of an instance
- the reclassification of an instance (in effect, the combination of the termination of a business object in one state and its creation in another state).

Within this schema, processes are considered to both respond to events and create events within a domain; all events are regarded as coalescing in some way to achieve the end result of some higher level process.

However, whatever the limitations of the literature it is possible to discern that the relationship between processes and events is a little subtler. What is missing is any concept of purpose or motivation; i.e. why is this process being executed?, how is the process responding to the event? Resolution of these issues is crucial if the description of the process is to capture the semantics of the domain.

The technique best equipped to capture the meaning of an activity appears to be Jacobson's use case model (Jacobson *et al*, 1992, 1994a, 1994b, 1994c, 1995, 1998). However, even with the benefit of various evolutions, Jacobson's definition of a use case (in a business system) as 'a sequence of transactions in a system whose task is to yield a result of measurable value to an individual actor of the business system' does not quite convey the necessary focus. Martin's definition of a value stream as 'an end-to-end collection of activities that has a clear reason for its existence - to deliver a result to a customer or end user' provides an alternative perspective (Martin, 1996). Both definitions provide for the delivery of a result, but little is revealed about the result except it that contains some value quotient (although Martin does not appear to even insist on this). The problems here are to define what is a result and what is value. To explore this definition it is necessary to return to the subject of Porter's value chain.

Martin contrasts a value stream with Porter's value chain which refers to a whole enterprise; value streams are simply a stream of activities within an enterprise that achieve a particular result. Martin notes that the ubiquity of attempts to define a process has rendered any attempt at a definition to be meaningless; in contrast, a value stream is defined unequivocally as 'a set of end-to-end activities that deliver particular results for a given customer (internal or external)'. The term 'end-to-end' is taken to convey the concept that a value stream will leave a business in an internally inconsistent state if it terminates prematurely; the only possibilities are that a value stream achieves its end result or it does not, there is no other option. Additionally, once a value stream commences execution it must terminate without branching to another value stream. These constraints (which incidentally echo the essential foundations of Ptech) may be used to augment the concept of Jacobson's use case model.



A use case may be redefined as 'a response to an event comprising an end-to-end collection of activities designed to deliver particular results of measurable value to a customer (or actor) of the system'. Within this framework, the activities constituting the use case are, in fact, the elusive business processes. By imposing the 'end-to-end' constraint, it is possible to interpret use cases as purposeful activities with a clearly defined motivation. Moreover, it is possible to extrapolate from the Halé approach that a use case may have only a single motivation and that the motivation may not be shared with other use cases. For each use case, the processes collaborate in some way to deliver the result or end-product forming the purpose of the use case. By further imposing 'separation of concern' and enforcing singularity of purpose for all levels of activity, it is now possible to approach the identification and definition of processes with some degree of confidence.

Analysis of some domain thus commences by identifying those environmental events to which the domain has to respond. Events may be detected as a result of executing routine operations or from surveillance of key environmental factors. Events may arise from the activities of customers, markets, suppliers, competitors, venture partners, governments and regulatory bodies. Events may be further classified as industrial, commercial, political, economic, media, social, ecological, financial and technological. Whatever the source of an event and however it is categorised, it may require a response from an organisation. The nature of that response provides the motivation for the subsequent activity; it could be argued that the response to any environmental event should result ultimately in delivery of a product that represents some increase in value to the customer. The response to an environmental event should be encapsulated in a use case with the motivation providing the purpose for use case. This principle may be extended to also include internal events and customers, although these would exist at the level of processes embedded in individual use cases. A use case is thus assigned to each class of environmental event being monitored by an organisation; the response to the event is defined by the specification associated with the use case that should describe a purposeful action delivering a discrete product. At this level of granularity, the use case should be supported by a narrative description identifying the actors, the

invocation events and the end-product, together with a description of the motivation for delivering the end-product. This refinement of use cases continues to resonate with Jacobson's original concept but also begins to address the issue of scalability that has proved to be problematic. The use case concept is transformed into a technique that may be used at the corporate level; the use case is redefined as a service that an organisation provides to its environment.

A contribution of the thesis is adopt Halé's Value-Analysis Diagrams to both extend and refine the concept of a use case. The approach to building the revised use case is as follows:

- detect an environmental event and analyse the implied behaviour
- determine what response is required from the organisation, i.e. what product or service is delivered to the customer
- determine what capability or resource is needed to achieve the response
- determine how the product or service is to be delivered.

---

Familiar naming conventions should be adopted to clarify and consolidate the existence of use cases and their associated capabilities. Typically, a capability is identified using a noun that may be qualified with an adjective or participle to indicate the state of the product, e.g. Confirmed Trade. The name for a use case should have two parts: the product to be delivered should be identified from the catalogue of capabilities; a verb in the imperative sense should describe the purpose of the use case. Therefore the name of a use case is formed from a verb and noun, e.g. Attract Client, Define Risk Limit, Execute Trade, Settle Corporate Action, and Raise Tax Claim. The selection of an appropriate verb is essential to identifying use cases; verbs such as Liaise, Manage, Monitor, or Perform are passive and do not indicate purpose, and, by implication, cannot deliver a product.

To provide a logical specification for a use case, it is necessary to study the collaboration required between the constituent processes to achieve its end-product (or purpose). From the preliminary description of the

use case, it is possible to identify both the events that launch and terminate the use case; from these events, all the processes constituting the use case may be identified and defined. As Halé notes, the creation, termination or change of state for a business object is an event that is the result of a process. Therefore to produce the model of a use case, it is necessary to show the cluster of processes and respective end results; this is analogous to modelling processes and their dependencies. A process is similar to a use case in that it may have only one end-result: the event that creates, terminates or changes the state a specific business object. The converse is also true: the creation, termination or realisation of a particular state for a business object may only be achieved by one process (another similarity with the use case concept). The invocation of a process is achieved by some combination of triggering events (which may be environmental events or the result of some other process). However, processes should not always be executed simply in response to the presence of some legitimate combination of triggering events; most processes require the imposition of some form of business rule which constrains the process, the constraint is referred to as a pre-condition which is defined as some restriction on the triggering events. The naming conventions for processes are identical to those for use cases, except that the verbs focus increasingly on the objects under transition. Moreover the business objects, upon which the processes act, may be hidden from the actor requesting the use case.

A business process may be described as follows:

- the purpose of the process
- the triggering events
- the pre-conditions constraining the invocation of the process by the event
- the post-condition identifying the terminating event and its relationship to the initial state, and specifying the state of the business object that achieves the purpose of the process
- the processing logic describes the tasks required to transform the pre-conditions into the post-condition.

This proposed method for describing processes raises some interesting points with respect to the association between use cases and their constituent processes. The process that creates the end event for a use case is unique to the use case and must therefore exhibit a broad alignment of purpose. All other processes may be (and frequently are) shared among use cases; for these processes, the purposes must remain independent of the purposes for any use case. Furthermore, use cases are abstract activities acting as a clustering device to provide motivation for executing processes. An environmental event may trigger the execution of multiple processes (indeed multiple instances of the same process) with use cases providing the motivation for each invocation.

Each use case is analysed by commencing with the terminating event and progressively working backwards to expose all the interim events until the original invocation events are encountered. With the exception of the originating events (the processes for which are specified in some other domain), each event requires the specification of an associated business process to constrain the conditions under which the event may be created. A process execution thus represents a 'thread of control' through a use case.

---

An event was defined previously as the creation, the termination, or reclassification of a business object. This definition provides the gateway between the business process and object models. From each event, it is possible to extract the underlying business object and its meaning, together with details of the state change pertaining to the event. Analysis of state changes provides the detail by which object behaviour is determined. Moreover, by considering dependencies between events, it is possible to elicit associations between business objects. Once these entities have been discovered, it is possible to revert to the familiar OO techniques of abstraction, encapsulation and inheritance to build a robust model that is resilient to change. In addition, the dissemination of processes and business objects throughout the organisation can be specified with the usual distribution matrices. Similarly, organisational, technical, cultural and social constraints may be assigned to processes and business objects.

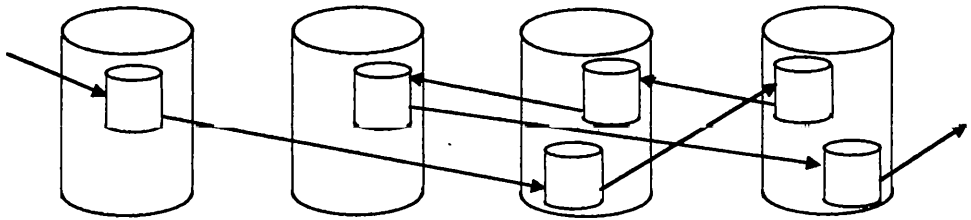
## **6.6 The Behavioural Barrier to Knowledge Creation**

From the perspective of knowledge creation, organisational dysfunctionality presents the most profound problem; it serves to both inhibit the expression of tacit knowledge and obscures an appropriate response to shifts in environmental behaviour. Organisational dysfunctionality can take many intricate, interdependent forms; in particular, political, social and cultural concerns may, if neglected or abused, contrive to introduce dissonance. These are outside the scope of this thesis. Of concern here is the dissonance that is introduced through the inappropriate arrangement of organisational behaviour.

Where an organisational configuration conforms to any form of hierarchical structure, the configuration is usually arranged according to some functional partitioning of the organisation. The functional partitions may be distributed geographically or replicated at remote locations. Furthermore, the functional partitions are usually based on a template drawn from the divisions enunciated in Porter's value chain (Porter, 1985). However, these divisions present the obstacles identified earlier with the subordination of process to function; with the new schema, a use case is subordinated to a function, and a function is subordinated to a value chain partition. A use case is thus subjected to two levels of fracture and dislocation. For a use case to represent an 'end-to-end' activity, it must be permitted to transcend both functional and 'value chain' division boundaries. However, in practice, the terms 'functional silos' or 'functional stovepipes' are used to reflect the fact that these partitions are constructed frequently with little regard for the requirements of the other partitions with which collaboration is necessary.

While this configuration may be appropriate for a unitary and mechanistic organisations operating exclusively in an orderly universe, problems arise when an organisation enters a complex universe in a bid to counter the influence of chaos. To survive in a complex universe, an organisation must be able to adapt endlessly and the existence of 'functional silos' militates against achieving this flexibility. The loss of flexibility nurtures an environment where opportunities are rife for redundancy, duplication and omission of business activities. In addition to the sub-optimal behaviour imposed by functional silos, there is

another consequence also injurious to the competitive vigour of an organisation. By dispersing activities artificially over some arbitrary structure, a level of complexity is added that conceals (or at least obscures) essential details of organisational behaviour. Any innovative revelation is thus likely to be influenced and constrained by the behaviour peculiar to an individual functional silo. The prospects of the innovation being implemented are diminished because due account may not have been taken of the requirements of collaborating silos; the organisation may thus miss a competitive opportunity or remain exposed to a threat.



**Figure 6-3 An illustration of how the execution of a use case may straddle multiple embedded functional silos.**

Another contribution to furthering the understanding of organisational behaviour is to propose how processes can be retrieved from 'functional silos' and reconfigured into use cases.

To begin it is necessary to ensure that use cases are not subordinated to functional silos, i.e. use cases become the dominant streams of behaviour drawn from the activities embedded deep in the functional silos. This is achieved by dismantling the functional silos to expose the inherent processes. The processes are extracted from each functional silo and reconstructed to implement the 'end-to-end' behaviour required of each use case. Following Porter's value chain analysis, use cases are either primary 'environment facing' activities or support activities that enable the organisation to operate in general and implement the primary processes in particular.

While this approach may appear to be eminently reasonable, the question remains of how precisely to reconstruct the remnants of the dismantled functional silos into effective and efficient use cases. For the answer, it is necessary to turn again to an event-driven analysis of the processes embedded in the functional silos. A use case is analysed to determine the end-product required to deliver the desired response to the invocation event(s). The event delivering the end product provides the mechanism for extracting the terminating process from the reservoir of processes populating the functional silos. The iterative application of the event-driven analysis techniques described earlier, identifies all the other processes required to define the desired behaviour of the use case. It is, of course, quite possible that the functional silos will not yield the necessary processes, or that some processes are not required to implement the use cases.

Use cases formed from deconstructed functional silos constitute the essential behaviour of the organisation and processes are the activities that implement this behaviour. When liberated from the mediating obstacle of functional silos, superfluous complexity is eliminated, exposing the essence of the threads of control and corresponding processes. This greater exposure clearly has profound implications for the cultivation of tacit knowledge. The descriptions of processes are distilled into their essential core; incidental 'noise' from organisational influence is eliminated. To optimise effectiveness and efficiency, organisations may embark on a restructuring arranged around logical threads of essential activities rather than arbitrary functions. Clearly the complexities associated with such a restructuring ensure that many obstacles have to be surmounted before the change is implemented. However, the provision of a precise description for the desired organisational behaviour establishes a robust bedrock from which to debate the obstacles. Under the new arrangement, an organisational unit is formed from a cohesive group collaborating with a single motivation, i.e. to achieve the purpose of a use case.

## **6.7 The Role of Reconstructed Processes in the Knowledge Creation Cycle**

Reconstructing processes to form 'end-to-end' patterns of behaviour eliminates the interference generated from the arbitrary clustering of processes into functional silos. The codification conventions may achieve a more direct association with the semantics of the domain under investigation, i.e. the use case. As Boisot reports, 'it is only when the tacit coefficient has been further reduced by additional efforts at codification that the new knowledge can become intelligible to a wider professional audience and hence diffusible to the community at large'. By reducing the demands of a dissonant cognitive investment and offering rewards more promptly, significant barriers to reducing the tacit coefficient are purged progressively from the analysis process.

A reduction in anomalous cognitive investments serves to illuminate those events defining the behaviour of the domain. It is essential to isolate and interpret these events as they emit the signals that may foreshadow shifting organisational or environmental behaviour. Reconstructed processes provide the conceptual framework from which organisations may self-organise by spontaneously forming special interest groups and coalitions focussing on the efficacy of process clusters. This endeavour is somewhat thwarted by organisations arranged around functional silos.

To conclude this chapter, it is argued that the greater clarity provided for the definitions of organisational behaviour in either an orderly or complex universe eases the knowledge creation cycle. It is assumed that organisational behaviour in a universe of chaos defies all attempts at analysis.

The departures into cognitive theory included in the thesis contend that the knowledge creation cycle is invoked when the domain expert detects a shift in the patterns of behaviour of either the organisation or its external environment. The new knowledge assets seek to either explain or respond to the shift, and it is through attempting to formulate a response that requirements for change evolve.



The arguments presented in this chapter have sought to establish the importance of offering a conceptually consistent representation of organisational behaviour. The contention is that this has been achieved by asserting that at all levels of granularity, invocations of behaviour are responding to events and also creating events that serve their purpose. It is claimed that unitary structures of organisation contrive to conceal and distort expressions of behaviour, while organic configurations liberate behaviour from the constraints of organisational convention.

Patterns have also emerged as an important concept in apprehending and interpreting organisational behaviour. It appears that the domain expert does not observe behaviour as isolated invocations of activity. Rather behaviour is observed as a complex pattern of activities that collaborate to respond to an event, with the process emerging as the primary unit of activity. The knowledge creation cycle is invoked when a shift in the pattern of processes raises the possibility that the response may no longer be appropriate to the event. The role of behaviour patterns as the foundation for the broader analysis patterns is discussed in the next chapter.

## Chapter 7

### 7 The Contribution of Analysis Patterns to Knowledge Creation

#### 7.1 The Case for Patterns

Many authoritative sources are exploring the existence of patterns within the context of human cognition and organisational behaviour: Mintzberg (1994), Kemeny (Kemeny *et al*, 1994), Senge (1994), Nonaka and Takeuchi (1995), Boisot (1995) and Claxton (1997) all refer to patterns explicitly or implicitly through discussions of intuition, mental models and domain schema.

The ubiquity and utility of patterns have, over the last few years, come to the attention of the IT community. In particular, two seminal works by Alexander (Alexander *et al*, 1977; Alexander, 1979) present the theory of applying archetypes to the design of houses, streets and communities; at the core of this theory is the proposition of the existence of 'languages' that allow individuals to articulate and communicate an infinite variety of designs within a formal system which lends them coherence. Patterns are considered to represent the units of this language and provide answers to design problems. Each pattern definition consists of a statement of the problem to be addressed, a discussion of the problem using an illustration, and a solution.

The significance of such a proposition is not lost on the broader IT community with the patterns movement emerging in the early 1990s. It should be noted that Alexander is something of a controversial figure within the domains of architectural design and construction. Moreover, Fowler notes that many of the pioneers of the patterns movement were exploring patterns without the benefit of any input from Alexander's body of literature (Fowler, 1997). Nonetheless, Alexander provides some invaluable conceptual touchstones for using patterns to design systems. It will be argued later that the analysis of IT systems and business systems may both be enhanced through the judicious use of patterns.

Alexander's pattern taxonomy is concerned entirely with architectural design and construction. However, many of Alexander's assertions may be abstracted to formulate a coherent set of general principles describing pattern theory. At the most abstract level, it is plausible to assert that any domain of concern may be described by a set of elements and a set of relationships connecting those elements. Moreover, the 'structure' of the domain may be considered to consist of patterns of relationships between the elements. These patterns are different each time they occur, but nonetheless exhibit some degree of self-similarity. Thus for a complex domain it is possible to conceive of a perpetual repetition of patterns characterised by an endless variety of features. From the theory of pattern languages (Alexander *et al*, 1977), it is claimed further that 'each pattern depends on both the smaller patterns it contains, and the larger patterns within which it is contained'. The pattern language is formed from the network of connections between patterns that contribute to the resolution of each individual pattern. The connections between the patterns are, therefore, almost as important as the patterns themselves in defining a pattern language. Alexander argues that every human endeavour and achievement is based on the exploitation and manipulation of some pattern language that is held in the mind of the individual. Furthermore, appreciation of a system is gained through an unconscious apprehension of the embedded patterns that inform speculative contemplation of that system. There now appear to be two pattern languages in existence: the one that is actually exhibited by the system and the one that is devised by an individual in an attempt to understand the system; and an incontrovertible objective for any approach to knowledge creation is to strive for the two pattern language schema to pursue convergent trajectories.

It is argued throughout the thesis that many business initiatives fail (in particular, the production of IT systems) because the techniques available for apprehending, eliciting and expressing requirements for business change do not reflect the realities of the climate in which the organisation interacts with its environment. It is claimed that the requirements with the potential to actually confer economic value on the organisation emerge from a climate of sustainable knowledge creation. Knowledge creation is sensitive to the symbolic conventions selected to elicit and express knowledge. However, even if

abstraction and codification conventions are available to provide a symbolic convention capable of giving coherent expression to the tacit knowledge held by the expert, there is now a suspicion that such a convention is alone insufficient to sustain knowledge creation. The ubiquity of patterns both as manifestations of a domain and as a cognitive device presents a persuasive argument for their explicit inclusion in a credible symbolic convention.

## **7.2 Pattern languages as an extended codification convention**

From Boisot's hypotheses on the epistemological space (E-Space), it is claimed that the purpose of a codification convention is to allow a domain expert to differentiate between concepts. Alexander attributes precisely this property to patterns. It is thus possible to conceive of pattern languages as forming some 'higher order' codification convention and to embed them firmly in the E-space. The concept of 'higher-order' knowledge may be developed further to distinguish between knowledge that emerges through the intentional use of patterns and that which emerges through some serendipitous speculation. This is not to underestimate the value of serendipitous knowledge, but to argue that its value may only be fully exploited once it is embedded in some configuration of patterns that yields both context and meaning. Without the guiding influence of a pattern language to capture the essence of a domain, it is more likely that any expression of knowledge will be more primitive and less interesting, and the knowledge creation process will become more complex and less certain. By contrast, a pattern language will confer a richer context on an item of knowledge and thus enable a more elaborate expression of the meaning and significance of the knowledge asset. This is of particular significance where knowledge is generated from detection of a weak signal in the environment. By contributing to a more elaborate expression of knowledge, a pattern language can contribute to both phases of Boisot's knowledge creation cycle, i.e. the value creation phase and the value exploitation phase.

A contribution of the thesis is to propose a synthesis between Alexander's pattern language and Boisot's SLC (Social Learning Cycle). Briefly, the value creation phase consists of the processes of scanning and

problem-solving followed by abstraction, while the value exploitation phase consists of the diffusion, absorption and impacting of knowledge. The contribution of pattern languages to the activities defining each phase is described below:

- **Scanning** describes the process by which individual (or small group) reflection transforms an item of diffused knowledge into 'singular and idiosyncratic' patterns. A pattern language contributes to scanning by offering a variety of patterns that may stimulate abstract reflection and indicate early opportunities for further speculation; scanning may thus be elevated to a more focused and less hesitant process of individual cognitive development.
- **Problem-solving** develops patterns yielded by scanning so they acquire a 'definite form and contour.' A pattern language provides a variety of templates with defined form and contour. By exploring the divergence between these templates and the concepts describing the problem under consideration, the pattern language may be extended. This activity is key to the knowledge creation process. By defining the form and contour, the codification is extended to include a new concept within the pattern language describing the domain. The implication here is that inclusion of a new concept may be the result of manipulating the pattern language to define new patterns or relationships between patterns.
- Boisot defines **abstraction** to be the act of extending the range of useful applications and generality that may apply to a newly codified pattern. This is the process whereby patterns are compared to determine whether any shared commonality of form or contour might suggest an opportunity for combining the patterns and thus simplifying the pattern language.
- **Diffusion** is the process by which knowledge is disseminated throughout a larger population. Pattern languages contribute to diffusion by conferring enhanced meaning on the knowledge presented to a less expert audience. The absence of any context isolates the knowledge and deprives the audience of the reassurance of any touchstones to provide a conceptual foundation; as Boisot notes, any such

disincentive will deter the audience from making the necessary cognitive investment to acquire the knowledge.

- **Absorption** internalises any newly created knowledge into a tacit state through repeated use. In this state, knowledge becomes embedded in mental models and thus, by common consent, becomes unreliable, elusive and difficult to express as the knowledge descends into the more obscure realms of cognitive activity. With this activity, the influence of pattern languages becomes somewhat tenuous. However, a pattern language may have some correspondence with the unconscious constructs devised for the mental model and better guide the descending cognitive trajectory until it disappears completely under the weight of abstract speculation.
- **Impacting** is the term coined by Boisot for new knowledge becoming embedded in concrete practices and physical systems. Boisot reasons that finding a new physical system confirms the viability of the knowledge and, if successful, increases its prospects for survival. Conversely, a failure to find a system suitable for embedding knowledge may limit its usage and any opportunities it presents for further learning. In this activity, it appears that pattern languages may contribute to identifying candidate systems by specifying the required form and content of a system within which the knowledge may be absorbed.

Boisot emphasises the existence of myriad SLC shapes that may be used to chart a trajectory through the I-space, many of which may be fragmentary and transient and not engage every stage of the value creation and value exploitation phases of the knowledge creation cycle. Moreover, the progression of any trajectory is dependent partially on the synergy between emergent knowledge and that represented by prior cognitive investments. Progression is likely to be accelerated where there is synergy, otherwise progress is likely to be blocked or re-routed to another trajectory through the I-space. As pattern languages are influential in all knowledge creation activities, it appears that they have much to contribute in extricating synergy from prior cognitive investments and thus avoid unnecessary blockages.

Boisot recognises that to justify a commitment to knowledge creation, an organisation must be compensated adequately for investment in the value creation phase, with the ability to drain knowledge of any economic value during the value exploitation phase. It was noted earlier that the ability to exploit knowledge is under endless assault from entropy that contrives to progressively deprive the knowledge of any vestige of economic value.

A contribution of the thesis is to argue that a pattern language may further contribute to knowledge creation by delaying the erosive influence of entropy as the emergent pattern languages are explored for new avenues of opportunity to exploit any latent value in the knowledge. The conjecture here is that an item of knowledge that has emerged without the benefit of a pattern language may be more transient and also expressed in a manner that is at once more primitive and uninteresting; i.e. opportunities for exploiting latent value are likely to remain concealed. The transformation is achieved not by prematurely abandoning knowledge that may possess latent value, but by exploring and adapting pattern languages to drive out any opportunities to exploit the latent value of explicit knowledge. With the support of a pattern language, analysis is targeting the expertise that defines unexpected, unintended and counter-intuitive behaviour and structure for the domain, i.e. the unsuspected opportunities to exploit value. Much of the organisational knowledge creation literature identifies this as precisely the knowledge required to acquire competitive vibrancy and offset the effects of erosion on the economic value of the existing reservoir of knowledge.

The case for patterns is that unless the use of pattern languages are factored into analysis and design techniques, results will be produced that have not absorbed the full quotient of expertise that is available as explicit knowledge. Without a pattern language, expressions of knowledge are likely to be diminished and uninteresting with a consequent loss of utility and opportunity to reward the investment made in acquiring the knowledge. Conversely, pattern languages contribute to a more complete and diverse elicitation, expression and implementation of expert knowledge. The greater depth and diversity of

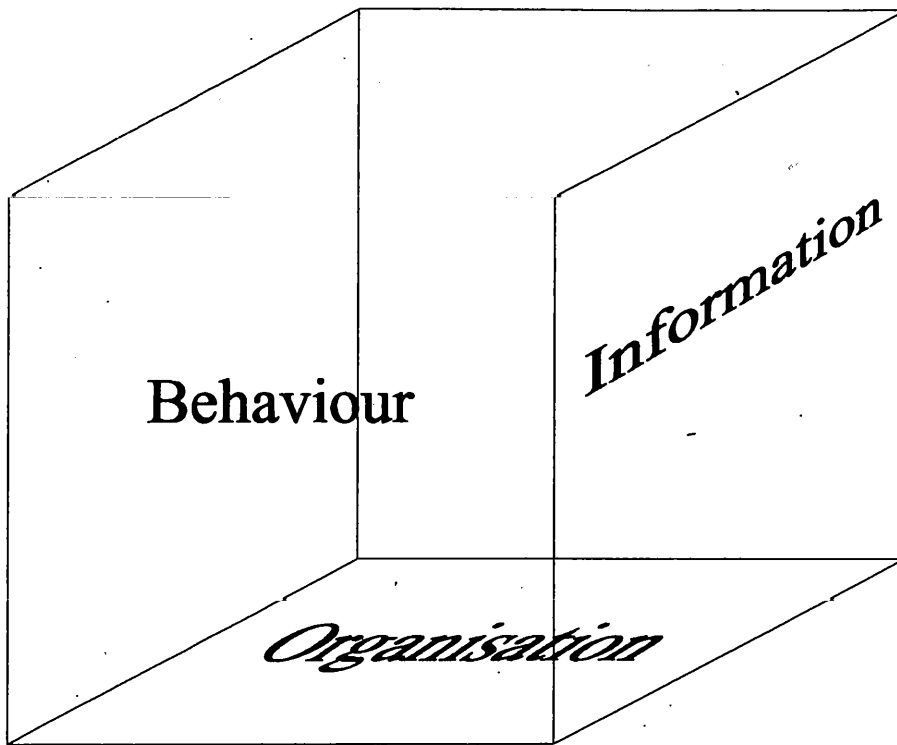
knowledge will also better equip the organisation to increase the 'bandwidth' of opportunities available to exploit the economic value of explicit knowledge. Furthermore, pattern languages provide a cognitive foundation for stimulating the generation of tacit knowledge and thus the invocation of future cycles of knowledge creation.

### **7.3 The Problems Associated with Pattern Languages**

Alexander warns that pattern languages offer no panacea for the difficulties to be surmounted in devising a 'timeless way of building', and so it is with business and IT systems where architectures are constructed largely on abstract concepts. It is observed that although patterns may appear simple once they are articulated, their discovery or invention may be far from simple and require multiple iterations.

However, the recognition of patterns in business and IT systems may be made less complex by observance of the fundamental analysis principle of 'separation of concern'. Analysis methods at all points of the maturity spectrum are predicated on the common precept that a domain should be analysed by exploring orthogonal dimensions independently before attempting some form of synthesis to develop a cohesive expression of the domain. However to ignore the influence of organisational topology (an omission shared by many IT analysis methods) invites details of how behaviour and resource are distributed to become embedded in the definitions and interactions. The risks here are obvious and grave: if the topology changes, the definitions of behaviour and resource are destabilised. The principle of 'separation of concern' dictates that organisational topology should be factored out of definitions of behaviour and resource (an action not dissimilar to the process of normalisation (Codd, 1970)) to form a new dimension representing organisational topology. The three-dimensional domain space is consistent with the hierarchy of business elements that provide the foundation for Taylor's theory of convergent engineering (Taylor, 1995).





**Figure 7-1 The three orthogonal dimensions of a business system domain**

By detaching concerns of organisational topology from pattern languages defining behaviour and objects (and other resources for that matter), the obfuscating influence of functional silos or any other dissonant topology is diminished. That is not to claim an accelerated cognitive progression; the domain expert may well have made considerable cognitive investments in developing mental models that are profoundly influenced by the organisational topology. However, to neglect the orthogonality between these dimensions is to present a formidable cognitive barrier to constructing pattern languages for any dimension. As the conceptual foundations for constructing the pattern language are undermined and destabilised, the phenomena of duplicated, redundant and omitted patterns is also likely to emerge. Following Boisot's reasoning, and indeed borrowing from Kuhn's theory of scientific revolutions, the domain expert is likely to pursue this thread of inquiry until the dissonance becomes intolerable. At this

point, further cognitive progress is blocked and the expert will either abandon the inquiry or shift to another trajectory of speculative discovery and invention. The analyst can, of course, spare the domain expert any unnecessary anguish by enforcing the discipline of 'separation of concern' through due observation of domain orthogonality.

Any trajectory in Boisot's E-space that aspires to acquire a 'higher-order' knowledge must therefore first strive to develop orthogonal pattern languages for behaviour, objects and organisation, and then construct a synthesis between the patterns.

#### **7.4 Patterns of Analysis**

Much space is devoted in the thesis to arguing that analysis must focus on the behaviour of the organisation and its external environment. It is through this focus that the knowledge creation cycle yields requirements for change that enable the organisation to adapt and survive when negotiating with the competing forces of order and chaos. It is argued further in this chapter that the exploitation of patterns of behaviour contribute to ensuring that knowledge assets are neither lost nor deprived of their opportunity of making a full contribution to the organisation.

A major contribution of the thesis is to propose a series of 'analysis patterns' that enable both the analyst and domain expert to engage productively in the knowledge creation cycle. The term 'analysis patterns', as used in this thesis, is defined as the variety of generic patterns that should be used to analyse a domain. The emphasis on behaviour throughout the thesis dictates that behavioural patterns should form the conceptual core of analysis patterns. From the earlier chapters addressing analysis methods and business processes, it is argued that analysis of the behaviour of a business domain is concerned with constructing a logical expression of the activities required to achieve the purpose of the domain. Pattern languages may contribute to the analysis process by arranging configurations of individual acts into cohesive clusters of

collaborative activity directed at achieving some purpose. By considering pattern languages within the context a business domain, the recursive nature of patterns to which Alexander alludes is revealed. At some level of abstraction, the irreducible acts of individuals collude at progressively remote and expanding levels of organisational strata to contribute ultimately to the mission of the enterprise. Clearly, this highly idealised scenario neglects interference from a variety of cognitive, sociological and organisational barriers that may contrive to divert or block the objectives of some collaborative activity. However, this scenario does suggest that organisational behaviour is recursive and replicates essential features at all levels of granularity. This recursive quality of organisational behaviour was recognised by Edwards (1984) and later by Halé (1995). However, as noted earlier in this chapter, unless expressions of behaviour are developed independently of organisational topology, any indication of pattern languages or recursion is likely to remain concealed.

Recalling briefly the discussions on organisational learning, an individual embarks on some trajectory of knowledge creation when the signals emitted by an event indicate a shift in the normal pattern of behaviour defining the interrelationship between an organisation and its environment. The ability to detect such a shift depends on the ability of the expert to interpret the message(s) contained within the signal. The signals may be weak and transient, and the messages may be incomplete or contradictory. It is from this often unpromising terrain that the expert has to formulate an appropriate response. What is appropriate is equally problematic, and may again be subject to transience, incompleteness and contradiction. However, a central claim of this thesis is that a behavioural pattern at least allows the expert to explore how the organisation is interacting currently with its environment and to evaluate various options for realignment. Without a pattern language to provide a conceptual configuration, the expert is likely to flounder under a deluge of fragmentary and dispersed expressions of organisational behaviour.

What follows therefore is a series of analysis patterns designed to support the core pattern addressing organisational behaviour. The purpose of each pattern is described using the following attributes; problem, context and solution.

#### **7.4.1 The Behaviour Pattern**

**Problem** How is the analyst to recognise instances of organisational behaviour and how are these instances to be described?

**Context** Analysts and domain experts are required to determine whether some activity constitutes significant organisational behaviour. A failure to define techniques that identify and describe organisational behaviour consistently and verifiably has proven to be a debilitating weakness in all threads of mainstream analysis methods. Unless the analyst is confident that organisational behaviour can be expressed according to a consistent and sustainable technique, the analysis process is undermined and cognitive developments are likely to descend into myriad unverifiable, inconsistent and contradictory conjectures. Analysis is aborted and a solution is embarked upon that can only draw on a problem domain that is incomplete and riddled with anomalies. Requirements for business change are thus either effectively concealed or may only surface in some distorted form. Certainly, any knowledge creation spiral is likely to be effectively blocked, and the ability to acquire future competitive leverage may become seriously compromised.

**Solution** The analyst and domain expert must be able to detect and interpret all instances of activity in all modes of behaviour. To define the pattern, it is necessary to return to hypotheses developed by Halé (1995), and adapt the concepts such that they are abstracted and thus apply to all behavioural modes. If activity is considered to be an expression of behaviour, the

activity is then defined as a set of operations, performed by agents (people or automats), that is supported by logistical resources. From the analysis perspective, a logical activity is an abstract description of an activity as a set of concepts, execution rules, events, preconditions and a post-condition. Design is concerned with devising a physical activity that is a particular implementation of a set of logical processes. A key constraint is that, at all levels of granularity, activities are purposeful and that purpose is achieved by the outcome of the activity. The outcome is achieved through delivery of an end-product; each activity may have only one end-product and an end-product may only be delivered by one activity. To deliver this end-product, an activity uses resources that were the end-products of other activities, i.e. the end-product of one activity is a resource for another activity. An activity may require multiple resources from a variety of other activities.

Halé draws on discussions of capability (Stalk *et al*, 1992) to refine further the definition of purpose. This discussion can be adapted to also refine the definition of the behaviour pattern. The essence of the debate is that an organisational activity should be concerned with not merely delivering an end-product, but also with providing a capability for the customer. From Halé, a capability may be defined as a set of resources or products that are required in a particular state to enable the user to achieve its objectives; a user may take many forms and be internal or external to the organisation exhibiting the behaviour. Halé builds on Grant's earlier definition of capability as 'the capacity for a team of resources to perform some task or activity' (Grant, 1991), by insisting that resources or products acquire a particular state before a capability is achieved. The concept of value is associated intimately with capability: drawing from Halé, value is defined as that property of an end-product which expresses a capability provided to its user. Thus an activity at any level of granularity should be analysed by first considering how the capability it provides fulfills its purpose. Not only are capabilities necessary to detect and interpret organisational activities, they have also proved influential in understanding the dynamic nature of strategy (Teece *et al*, 1991) and

knowledge creation (Leonard-Barton, 1992). In particular, Leonard-Barton defines a core capability to be 'the knowledge set that distinguishes and provides competitive advantage'. It is thus imperative that, at all levels of granularity, an organisational activity delivers an end-product providing a capability to its user and thus augments the organisation's reservoir of knowledge. To neglect capability is to invite competitive decline and marginalise the utility and contribution of knowledge.

The behaviour pattern exploits the holographic form of purposeful activity by providing a definition that applies at all levels of organisational endeavour. Thus organisational behaviour may be viewed as an endlessly shifting ensemble of recursive activities that collaborate to form an adaptive complex system transcending atomic activities, processes, functions, operational units, regional units and even the organisation itself.

#### **7.4.2 The Behaviour Typology Pattern**

**Problem** The vocabulary available for describing types of activity is limited and overused to the extent that individual terms now represent meaningless concepts. Examples of such terms would include: task, action, process, procedure, function, value chain and value stream. The conceptual integrity of any expression of behaviour is thus confronted by the ever present threat of ambiguity.

**Context** Embryonic attempts at knowledge creation are usually an abstract reflection of organisational behaviour leading to speculative contemplation of resource and organisation. Unless behaviour can be partitioned into categories that guide cognitive development, knowledge creation is likely to be blocked by conventions that introduce ambiguity and uncertainty when clarity of concept is most needed. A pattern is required to define a typology of behaviour that is consistent for activities at all levels of granularity.

**Solution**     The Behaviour Typology pattern declares that there two types of organisational behaviour:

- **Core activities** deliver the core capabilities required by the organisation to achieve its strategic intent. This definition is clearly consistent with Porter's concept of primary activities, but does not conform particularly well with support activities. Halé addresses this by querying whether or not support activities form that part of the primary activities 'delivering the physical resources required by the organisation in the agreed form and the best possible terms and conditions'. Porter's value chain is thus transformed into a collaborative web of core (primary) activities that are responsible for their own infrastructure and resource management. If some activity does produce a capability that is still not recognised as a core activity, Halé suggests that the definition of strategic intention is evaluated to confirm the contribution of the activity.
- **Management activities** provide the enabling environment necessary to foster the generation and delivery of capabilities. To deliver this mandate, management activities must: determine the overall direction of the organisation; managing the capabilities required to sustain the direction, managing the activities implemented to deliver the capabilities. Halé comments that these classes of management activity do not imply any hierarchy, but should be viewed as illustrating the collaboration required between strategic, tactical and operational activities.

Halé subdivides management activities into a lower level typology:

- The Direction Management activity formulates the strategic intention, identifies the required core activities and monitors their performance by evaluating their contribution to strategic objectives.
- A Capability Management activity creates one of the core activities identified by the Direction Management activity, and manages its contribution to strategic objectives.
- An Operational Management activity uses and manages the capabilities (the resources) made available by the Capability Management activity in order to contribute to fulfillment of the strategic intention.

### **7.4.3 The Environmental Behaviour Pattern**

**Problem** Within the maelstrom of corporate life, it is often difficult for the analyst to locate those activities that will yield knowledge assets of genuine benefit to the organisation.

**Context** The domain expert is surrounded on a daily basis by a multitude of expressions of organisational behaviour. To be able to embark on any trajectory of knowledge creation that has even the remotest prospect of producing anything of value, the expert must be able to distinguish between behaviour which is merely noisy and incidental, and activities delivering capabilities to customers and the broader environment. To achieve this, the domain expert may need guidance on where to look for meaningful business activity.

**Solution** It has been reiterated throughout the thesis that a knowledge creation cycle germinates from a domain expert detecting some shift in a signal emitted by an environmental event. Knowledge is created when the expert is able to decipher the signal and devise some response that adapts the behaviour of the organisation to its environment. It can be inferred from this reasoning that all core activities collaborate to respond ultimately to environmental



events; consequently, environmental events are the primary source of all business activity and the knowledge that flows from resolving disturbances in the signals emitted from events.

Although this assertion may move the pattern forward a little, it is still not a conclusive resolution to identifying core activity; i.e. the expert is now confronted with a new conundrum, where are the events to be found? To resolve this conundrum and complete the definition of the pattern, it is necessary to turn to an inventory of organisational stakeholders; in addition to the familiar list of employee, customer, supplier and shareholder, it might also be prudent to include competitor, venture partner, regulatory body, trade union, labour group, national or local government, media, pressure or lobby group, market, exchange, community and society. Any of these entities, whether acting as individuals or groups, may create an event to which the organisation may be compelled to respond. Moreover, the organisation may be required to adapt its response if the event emits a signal that displays any disturbance.

---

Core activities are discovered by first identifying stakeholder events and then devising a response by the organisation. Management activities commence by defining the core activity and proceed by providing the infrastructure to implement it.

#### **7.4.4 The Organisational Behaviour Pattern**

**Problem** A primary challenge for the analyst is to describe how an organisation intends to respond to some shift in the behaviour of the organisation or its external environment.

**Context** Earlier approaches to modelling organisational behaviour depend largely on eliciting activities from functional partitions that usually reflect some corporate topology. Little

attention is paid to purpose and motivation. Analysis is concerned more with defining the sequence, selection and iteration of individual tasks, although this focus could also become confused where the phenomena invoking and terminating activities are uncertain. Consequently, the domain expert has difficulty matching instances of organisational behaviour to environmental events. If the matching process is unreliable, any uncertainty is likely to escalate and cascade into dependent cognitive endeavour.

**Solution** The Organisational Behaviour pattern has its foundations in Jacobson's use case (Jacobson *et al*, 1992). The concept of a use case has undergone many shifts of nuance and meaning since its inception. The definition that is most pertinent to the thesis is 'a sequence of transactions in a system whose task is to yield a result of measurable value to an individual actor of the business' where an actor is defined as 'one or a set of roles that someone or something in the environment can play in relation to the business' (Jacobson *et al*, 1994). This definition bears a resemblance to almost all definitions of organisational activity that appear in the literature; in particular, it resembles almost any definition of a business process and Martin's definition of a value stream (Martin, 1996). However, the utility of a use case can be transformed by the inclusion of two simple expedients:

- Each use case must be attached to an environmental event. A use case is thus the organisational behaviour that constitutes the response to the event and delivers a capability to the actor; it might be argued that all use cases should ultimately deliver a capability to the customer (the primary actor). Furthermore, an event type might invoke multiple use cases, but a use case may only invoke a single event. The provision of the capability for the actor yields the motivation of the activity and thus gives the purpose for the use case. The purpose is achieved once the end-product or goal is available; this constraint on the definition of a use case exploits the Behaviour Pattern. The Behaviour

Typology may also be utilised by recognising that use cases may be either core or management use cases.

- The delivery of the end-product provides the basis for further constraining the use case. Borrowing from Martin's definition of a value stream, a use case may be similarly defined as an 'end-to-end' collection of activities that collaborate to create value by delivering a capability to an actor. The term 'end-to-end' is significant in defining a use case in that it insists a use case terminates, either successfully or unsuccessfully, without passing the thread of control to another use case. Moreover, upon termination, the use case must leave the organisation in a consistent state. The 'end-to-end' condition offers no option other than for a use case to either achieve its goal or fail; and for either route the organisation is left in an orderly state. The condition borrows heavily from OO where highly modular objects are fully responsible for the conduct of their own behaviour and should remain independent (unaware) of other objects. Indeed, Jacobson provides 'control' objects to manage the behaviour of use cases.

The definition of the use case is confined to a brief description of the motivation for the behaviour and how resources will be transformed into an end-product (or goal) that delivers a capability to the actor. A noteworthy constraint on the use case is that its description should be independent of organisational structure (this is dealt with later by the Domain Synthesis pattern); use cases often represent high-level activities that can transcend traditional functional and organisational boundaries.

#### 7.4.5 The Process Pattern

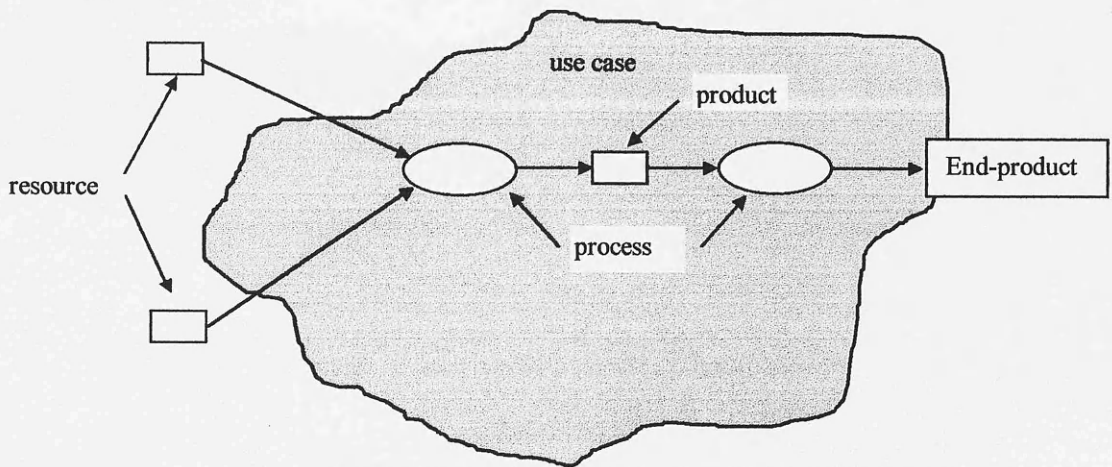
**Problem** A use case may require the execution of a complex web of individual acts before its purpose is achieved. The problem is to identify these acts and define their legitimate interdependencies.

**Context** When an expert has detected disturbance in an environmental signal, an imperative is created to formulate an appropriate response. Once the affected use case(s) has been identified, the expert has to explore the patterns of behaviour embedded in the use case and reassemble them such that they implement the desired response to the shifted environment.

**Solution** The term business process is reserved for those activities that fulfill the purpose of a use case; in this context, business processes are likely to be activities of a much finer granularity than is normally encountered in the literature. It is now recognised however, that a fine granularity of definition is necessary to confer crucial versatility on the specification of business activities (Jackson & Twaddle, 1997). The Behaviour Pattern is used again to define the shape of the business process, but is now refined by restricting the end-product to be a change of state of a concept contributing to the delivery of a capability. By sharing the Behaviour pattern, it is possible to confer the same characteristics on each activity while varying the focus to achieve the required level of granularity. Use cases are distinguished from processes by the fact that processes collaborate through a web of state changes to create an end-product (another state change) that achieves the goal of the use case and delivers a capability to an actor. It should be noted that decomposition is not required when building process models; processes are immediately atomic and are clustered to highlight patterns of activity and support cognitive development. Further descriptions of processes may be located in the earlier material on Business Processes.

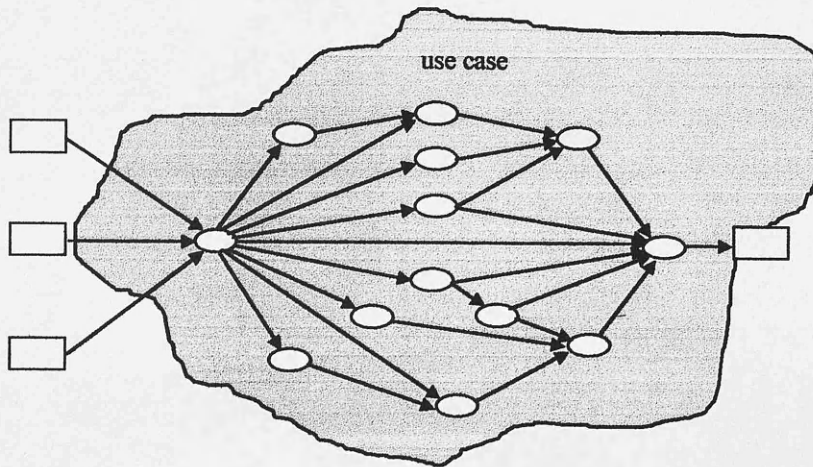
By insisting that a process may only terminate (or fail) upon invocation of the event signifying a change of state for the concept underlying the process, it emerges how a web of processes might be represented.

- For a trivial use case, the processing logic may be described by a dependency between two processes, with events signifying the availability of the resources, and the creation of the product and end-product respectively.



**Figure 7-2 A process pattern for a trivial use case**

- For a non-trivial use case, the web of complexity is expressed through direct dependencies between processes which are supplemented by layers of indirection representing intermediate events that must be invoked to satisfy the pre-conditions of the dependent process.



**Figure 7-3 A ‘diamond-shaped’ process pattern for a non-trivial use case.**

The patterns of behaviour described earlier may be used to create use case and process patterns that can be instantiated to express the functionality of a domain. The expert might therefore expect a domain to be described from an ensemble of holographic, self-similar patterns that define the collaborative activity required for the domain to achieve its purpose. Moreover, the expert might reasonably expect the functionality of the domain to be extended by replicating some cluster of patterns in a slightly modified form. Finally, the expert should not be surprised if patterns from other domains can be utilised in some slightly modified form.

#### **7.4.6 Patterns of Objects**

As the development of patterns theory is drawn largely from contributions submitted by the OO community, it is probably not surprising that objects provide the primary focus of the literature (Gamma *et al*, 1995; Coplien & Schmidt, 1995; Buschmann *et al*, 1996; Partridge, 1996; Fowler, 1997; Coad, North & Mayfield, 1995). The only notable exception is Hay who seeks to apply patterns to entity-

relationship models (Hay, 1996). With the exception of Coad *et al*, Fowler and Hay, the formulation of object-based patterns is concerned entirely with designing the solution domain.

Coad *et al* made the earliest contribution to analysis patterns with the approach and narration very much in the style of the previous Coad and Yourdon offerings on OO analysis (Coad & Yourdon, 1990 & 1991). There is an almost total reliance on the analyst's powers of inference to discover the domain objects from what can only be described as perfunctory dialogues with the expert. From these dialogues, the analyst constructs domain models of a dozen or so objects; it should be noted that these techniques would be singularly inappropriate for non-trivial systems. In this respect, the Coad approach has much in common with many of the 'naive' OO analysis methods. However, Coad's contribution should not be discounted as the effort is, to some extent, redeemed by the inclusion of a strategies and patterns handbook that is of interest. The strategy for building object models is standard fare: identifying system purpose and features; selecting objects; establishing responsibilities; working out dynamics with scenarios; discovering new strategies and patterns. However, Coad's contribution is noteworthy for its attempt to venture a set of patterns for building object models. A pattern is defined to be 'a template of objects with stereotypical responsibilities and interactions, where the template may be applied again and again'. Patterns are categorised into the following pattern families:

- The 'Collection-Worker' pattern is the fundamental pattern and provides the root for all other object-model patterns. The pattern represents a one-to-many relationship between a 'collection' object and a group of 'worker' objects.
- Transaction patterns describe the object structures necessary to define a generic transaction; the patterns include: actor-participant, participant-transaction, place-transaction, specific item-transaction, transaction-transaction line, transaction-subsequent transaction, transaction line item-subsequent transaction line item, item-line item, item-specific item and associate-other associate. While an experienced analyst would doubtless find much to criticise with Coad's transaction patterns, they are at least interesting in suggesting a way of discovering or inventing patterns. In fact, all but the

generalisation-specification associations are considered to be patterns that collaborate to form some expression of a transaction.

- Aggregate patterns describe various types of aggregation constructs: container-content, contain-container line item, group member, assembly-part, compound part-part, packet-packet component. Again an analyst may find fault, as the difference between some patterns is probably little more than nuance and depends on interpretation of the terms; the suspicion is that either these patterns overlap or are otherwise erroneous.
- Similar patterns are provided for planning and interaction domains.

Coad's categories should not be judged for their completeness or accuracy, but rather as an approach to classifying and discovering object patterns.

Hay's contribution is something of an oddity; it makes no reference to either patterns or OO literature and follows the Oracle notation and method (Barker, 1989). In common with all approaches that are oriented towards the data-driven perspective, Hay embarks on an odyssey of conjecture to construct entity models and infer the existence of embedded data patterns. Hay does, however, provide an interesting insight into the nature of patterns by commenting that groups of entities selected as patterns should have semantic content. Following Hay's comment, it is clear that a pattern is not merely an arbitrary cluster of objects that can be reused in another context, but that it should also have a qualitative dimension; i.e. a viable pattern should have meaning and purpose that can be discerned by the expert.

Fowler's approach to analysis patterns follows clearly in the tradition of the Gang Of Four (Gamma *et al*, 1995) and Coplien and Schmidt (1995). For Fowler, a pattern is defined as 'an idea that has been useful in one practical context and will probably be useful in others' and may be anything that has been discovered (or invented) from practical experience of a real project. Fowler's patterns fall into two categories:



- Analysis patterns are groups of concepts that represent a common construction in business modelling and may apply to multiple domains.
- Supporting patterns that describe how to implement analysis patterns.

Although Fowler adopts the Ptech notation (Edwards, 1984), the approach to analysis appears to be aligned more to the Booch method or Rumbaugh's OMT. Accordingly, there is more emphasis on modelling the conceptual structure of a domain rather than its behaviour, and this is where, it is claimed, the analysis patterns are to be found. Although the behaviour of objects is explored through the use of event diagrams, interaction diagrams and state diagrams, there is no sense in which object behaviour is examined to confirm whether or not it has delivered the end-product of a use case which, presumably, defines its domain. Moreover, the domain behaviour seems to be inferred from the structure of the object model and the responsibilities conferred on each individual object; i.e. function is following form, an inversion of what might be expected from any system that has aspirations of sustainability.

In common with Fowler's discourse, Partridge also adopts the Ptech notation to illustrate an object paradigm incorporating business object patterns. Partridge also shares Kay's concern with using patterns to capture the semantic content of some cluster of objects. At the level of individual objects, Partridge recognises that the simplest pattern exists when only two objects are involved (there are obvious echoes here of Coad's fundamental pattern) and acquire greater complexity for larger groups of objects. By simply considering two objects, Partridge identifies three main patterns of connection: distinct, overlapping and whole-part. When considering a larger group of individual objects, there may be combinations of distinct, overlapping and whole-part patterns that may apply also to the whole group. Moreover, distinct and overlapping patterns for individual objects may be inherited in opposite directions along the whole-part hierarchy; i.e. distinctness is inherited down the whole-part hierarchy and overlapping is inherited up the whole-part hierarchy. Partitioning is also used to identify patterns. Objects may be partitioned into distinct parts of the whole object by combining the whole-part and distinct patterns into a partition pattern; moreover, partitions may be inherited down the whole-part hierarchy.

Finally, logical dependency can be created between overlapping objects through intersection and fusion patterns.

Partridge continues by examining state hierarchy (spatio-temporal) patterns: state/sub-class hierarchy patterns, state/sub-state hierarchy patterns, distinct state patterns, partitioned state patterns and overlapping state patterns. Time-ordered temporal patterns are considered with three examples: simple state 'change' patterns, time sequence of state patterns (object level and class level sequences), and alternating state patterns.

Partridge also explores cardinality patterns of tuple classes. It is declared that cardinality, in object syntax, applies to occupied class place objects, i.e. to those objects that can occupy a class place. A cardinality pattern must have its lower and upper bounds specified, where a lower bound may take a value of optional or one, and the upper bound may take a value of one or many. Thus by allowing for the possible combinations of lower and upper bound values, four possible cardinality patterns emerge for the occupied class place:

- optional-to-one pattern
- one-to-one pattern
- optional-to multiple pattern
- one-to-multiple pattern.

To complete the material on patterns, Partridge examines a pattern for compacting classes; the motivation here is to simplify models by generalising classes. The pattern is implemented by placing a class higher up a super/sub-class hierarchy, thus eliminating the original, more specific, place classes. By exploiting shared abstractions, common concepts can be integrated, thus compacting the model without sacrificing its integrity. This pattern appears to be fulfilling one of Boisot's epistemological criteria of knowledge creation.

Partridge's exploration of analysis patterns is superior to those of both Coad *et al*, and Fowler. While Coad *et al* have at least attempted to define a theoretical foundation, Fowler has sought to establish few theoretical principles and has been content simply to illustrate extracts of models that have described certain phenomena in particular domains that may be of use elsewhere. This is not to argue that, in either case, the models are anomalous (this is actually impossible to establish), but simply to point out that they do not necessarily conform to any of the criteria proposed by Alexander, Gamma *et al*, or Coplien and Schmidt. Partridge provides a catalogue of essential analysis patterns and has succeeded in producing a more extensive repertoire based on sounder theoretical principles.

However, it appears that all the approaches to developing object patterns are missing the most fundamental object pattern; indeed, a pattern that follows from Halé, i.e. the Conceptual Domain pattern.

### **The Conceptual Domain Pattern**

---

**Problem** Many analysis methods are totally reliant on a grammatical parsing of some requirements statement followed by an odyssey of heuristic conjecture based on fragmentary domain knowledge. How is the analyst to verifiably confirm the existence and configuration of business objects for a domain?

**Context** The heuristic approach to the initial identification and configuration of business objects has, almost by default, become a de facto standard for the mainstream analysis tradition. Yet this standard is also the source of many of the barriers that can block a knowledge creation effort. While some early progress may appear to be made in identifying the major (obvious) business objects, problems are soon encountered when analysis turns to the behaviour of those objects that require a more sophisticated level of reflection. Analysis of object behaviour is necessary both to yield a definition and reveal the interaction (and thus the

configuration) between objects. While it may be enticing to suppose that a brainstorming exercise can result in some definition of a domain, this definition is almost certain to be incomplete and unverified (this point is now acknowledged by its proponents (see (Bollin & Simone, 1997)). However, what appears not to be recognised is that business objects yielded by heuristical techniques are likely to produce a suspect foundation for a more formal analysis that is bound to follow.

For analysis of a domain to be effective, a technique must be devised that is consistent at all levels of granularity and verifiably confirms the existence and configuration of business objects defining a conceptual domain.

**Solution** Following Halé's description of a process, a Conceptual Domain is constituted of the configuration of business objects required to deliver the end-product of an activity domain and is constructed for both use cases and business processes. By focusing on essential business objects the Conceptual Domain pattern provides a credible and sustainable convention for specifying an abstraction and codification notation supporting advanced cognitive development.

The Concept Domain pattern is integrated with the Behaviour pattern through the exploration of events associated with the domain of a business activity. Following the Ptech method, an event is defined as a change in the state of a concept that may be:

- the creation of an instance
- the deletion of an instance
- the reclassification of an instance (Halé comments that reclassification is actually the termination of an object in one state and its creation in another state).

Crucially, an event is the result of a process. Furthermore, processes collaborate to create a web of events that culminate in achieving the end-goal (or purpose) of a use case. An event triggers at least one other process, with pre-conditions controlling the invocation of the triggered process. Events are either conditional or unconditional; conditional events require a process to evaluate at least one condition before determining whether to create the event or terminate with no outcome, whereas an unconditional event is created for each invocation of a particular process. It also holds for the Conceptual Domain pattern that an event can only be created by one process and a process can only create one event. An event occurs when a business object achieves a particular state, and should thus be accorded a name that expressly identifies the object and its acquired state, e.g. 'withdrawn order', 'settled deal'. By exploring the web of events associated with the implementation of a use case, the business objects exposed by those events become available to populate the conceptual domain. Similarly, the dependencies between events can be used to reveal the associations between objects and thus define the configuration of the conceptual domain.

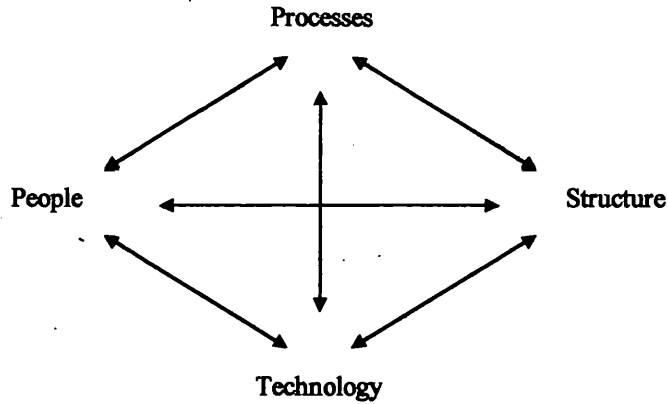
The Conceptual Domain pattern declares that the discovery and invention of classes and objects should not be a matter of conjecture, but should be derived verifiably from the web of resources and products required to support the collaboration of processes necessary to implement a use case. Object patterns are only given meaning and relevance when providing a conceptual foundation for an associated pattern of behaviour.

#### **7.4.7 Patterns of Organisation**

Organisations in all their manifestations provide another space where the analyst might expect a proliferation of patterns; James Coplien has written extensively in this area and is considered to be an

authority (Coplien & Schmidt, 1995). Coplien draws inspiration from the earlier works of Alexander (Alexander *et al*, 1977; Alexander, 1979) and notes, in particular, the holographic quality of patterns that enable complex behaviour to emerge from a set of simple patterns. Coplien claims for the pattern language that practitioners in 'highly productive organisations' have confirmed the effectiveness of each prescribed pattern. Coplien explores both organisational and process patterns; although the context is concerned broadly with developing application software, some of the patterns apply equally well in a broader arena. With respect to organisations, Coplien is particularly interested in the size, structure, orientation, operation and population of organisations. Coplien's process patterns are of less interest in a wider context, as they are concerned solely with the software engineering process. However, even with this limited horizon, it is possible to identify the necessity for a coherent architecture against which decisions may be reviewed. An architecture is formed from an integrated set of orthogonal dimensions that define the framework of a system; Domain Synthesis patterns define how orthogonal dimensions may be integrated and are examined next.

Leavitt's diamond (Leavitt, 1965) alerts the analyst to the fact that disruption to one dimension is likely to disturb the other dimensions defining the socio-technical system framework describing the organisation. Galliers points out that proponents of the systemic movement of strategic management add cultural issues to Leavitt's diamond (Galliers, 1995) and thus introduce further opportunity for disruption. However, Leavitt's diamond is of further interest because it espouses the fundamental principle of 'separation of concern'; a principle that is frequently violated and rarely implemented consistently and completely.



**Figure 7-4 Leavitt's diamond illustrating a socio-technical framework (Leavitt, 1965)**

In addition to illustrating the interdependency between the socio-technical dimensions of an organisational system, Leavitt's diamond may also be used to remind the analyst of the criticality of building orthogonal expressions for each dimension of a socio-technical system. Some mapping mechanism may be devised to associate the nodes within and between each dimension. To compromise the integrity of one dimension with material from another dimension is to risk the intrusion of an escalating cognitive barrier that may effectively block knowledge creation. Leavitt's diamond is therefore an expression of the fundamental principle.

### **The Domain Synthesis Pattern**

**Problem** Any non-trivial system is constituted from a multiplicity of interdependent dimensions that contrive to present a space of bewildering complexity. The problem for the analyst is to unravel that complexity and present an expression of the system that is coherent and precise.

**Context** When exploring a non-trivial domain, it is often tempting not to recognise the orthogonal dimensions defining that domain and reflect on an item of knowledge that draws material

from multiple dimensions. The temptation is introduced because knowledge in an embryonic form, i.e. tacit knowledge, is often immersed deeply in mental models that are messy, inconsistent and ill-structured (Norman, 1983). Unless orthogonality is observed, cognitive development is likely to be blocked as assumptions, values and beliefs continue to infiltrate evolving edifices of knowledge and yet remain concealed.

**Solution** The Domain Synthesis pattern declares that before a non-trivial system is examined, all the orthogonal dimensions for the system should be identified and some abstraction and codification convention established for each dimension. Thereafter, all expressions of new knowledge should strive to observe the integrity of the orthogonal dimensions. Finally, when each dimension is defined sufficiently by a web of entries, mappings should be attempted between nodes from respective dimensions. The mappings should form part of the various notation conventions and may be either simple or complex; a simple mapping merely denotes the existence of some association, while a complex mapping seeks to qualify or otherwise refine the definition of the correspondence.

## **7.5 The Case for Analysis Patterns - a reprise**

The case for analysis patterns represents the culmination of much of the discussion in the thesis. Clearly, though, the case for analysis patterns has been argued extensively elsewhere in IT literature so such a contribution to knowledge could hardly be considered to be original.

The contribution of this thesis, however, is to trace the need for analysis patterns back to the source of requirements for change. It is argued in the thesis that where an organisation is seeking to negotiate a trajectory between the competing forces of order and chaos, it is necessary to change or, more precisely, to adapt. The ability to sustain adaptation is only possible where an organisation has an endless commitment



to learning and knowledge creation. Abstraction and codification conventions, and the use of patterns are presented as pivotal contributors to the knowledge creation cycle. For it is not simply knowledge assets that yield requirements for change, it is also configurations of knowledge assets. Moreover, it is these configurations that have the potential to yield the greatest benefit to the organisation.

To simply extol the virtues of patterns as an analysis tool is, however, to rather miss the point. An expert is distinguished from a novice through access to a wider variety of patterns and an enhanced ability to manipulate those patterns. Cognitive theory asserts that it is patterns of behaviour, or rather shifts in these patterns, which first prompt the expert to contemplate the relationship between an organisation and its external environment. It would seem, therefore, that a primary objective of the analyst is to construct configurations of knowledge assets such that they resemble patterns of behaviour exhibited by the organisation and its external environment. If an analysis method is to support sustainable knowledge creation, it must be augmented by a variety of patterns that establish organisational behaviour as the focus. Thus, although patterns have made a worthy contribution to our understanding of the analysis process, it is only when they assume a behavioural focus that they are better placed to yield requirements of sustainable benefit to the organisation.

Erosion in the utility and scarcity of knowledge is a topic that has surfaced in the thesis. It is argued that patterns allow knowledge assets to be placed into complex configurations and thus provide a rich context from which to reflect and speculate on the significance of each asset. This complex web thus improves the prospects for any latent value in the knowledge asset to be revealed more readily. In this way, patterns offer some resistance to the erosion in the value of the knowledge asset. Another key consideration is the role of the analyst.

Hitherto, the analyst has been assigned traditionally to individual projects where contributions to knowledge creation were discrete and fragmentary. Such a sporadic involvement cannot sustain a continuous knowledge creation cycle. Assaults on the value of contemporary knowledge require that the

domain expert must be engaged continuously in complementary processes of externalization and internalisation, and must therefore be in regular dialogue with the analyst. A case can now be made for reconstructing the role of the analyst into that of custodian for the corporate knowledge base. The analyst is therefore no longer simply concerned with eliciting requirements for change in some discrete way, but now fashions knowledge assets such that they yield requirements enabling the organisation to adapt in a complex universe. 'Behaviour-accented' patterns of analysis are key to this endeavour.

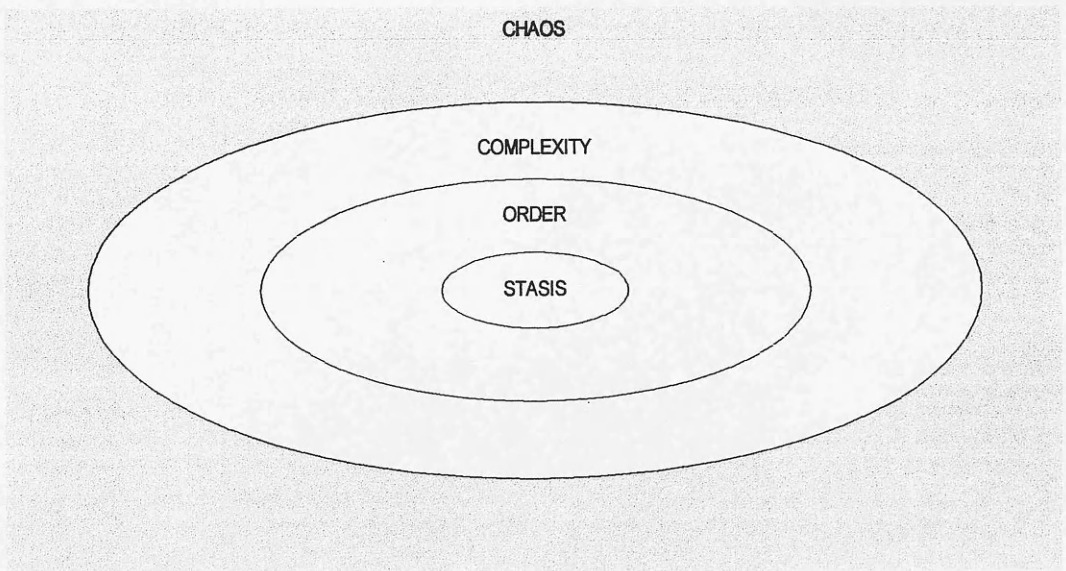
Although the future is unknown, it is not necessarily unknowable. In the universe of complexity between the competing influences of order and chaos, shifts in patterns of behaviour are difficult to predict but are, nonetheless, occasionally predictable. It is this possibility that provides the justification for enquiring into the nature of analysis.

## Chapter 8

### 8 Conclusion

#### 8.1 The pre-eminent cause of system failure

An attempt has been made with the thesis to argue that changes to IT and business systems fail because the various prescriptions for change neglect the true nature of an organisation. Figure 9.1 illustrates four classes of behaviour that may be available to an organisation at any point in time.



**Figure 8-1 Four classes of organisational behaviour (Battram, 1998)**

Stasis depicts a class of behaviour where the organisation is formed from unchanging patterns of cells that prohibit exchanges of information; i.e. the organisation stagnates and is effectively defunct.

Order reflects the behaviour of an organisation where the cells shift between a set of two or more states in an eventually repeating pattern. The diagnosis here is that the organisation is 'complacent and

unresponsive – repeating patterns over and over again’. Opportunities for exchanges of information are considered to be limited. An ordered organisation is considered to be ‘one that is not adapting, not responding to change’.

Complexity represents behaviour that displays complex patterns with elements of order and disorder that evolve and change. Organisations exhibiting the features of complexity are considered to be effective and creative, where information exchanges are flexible enough to transmit messages while stable enough to support message structures.

Chaotic systems permit cells to grow and die with no discernible pattern. Information cannot be transmitted; neither message structure nor message transmission can be supported. This is represented as an organisation ‘at war with itself’ where no useful work is done; i.e. the organisation collapses into disintegration.

Complexity is considered to ‘exist on the “edge of chaos”, poised between order and chaos’. To inhabit the ‘edge of chaos’, an organisation must behave as a complex adaptive system; the organisation must display ‘emergent patterns of creative, adaptive behaviour’. As a complex adaptive system, an organisation will respond to its environment, ‘changing it in the process and be changed by it’.

## **8.2 The difficulty with requirements**

Much of the literature concentrates on the capture and expression of requirements when the behaviour of an organisation is in a state of stasis or order. The thesis proposes a new approach to requirements engineering for organisations exhibiting the behaviour of a complex adaptive system. It is assumed that when the behaviour of an organisation is classified as chaotic, the formulation of requirements is impossible.

The formulation of requirements for a complex adaptive system, however, poses new challenges for the analyst and domain expert. A requirement can no longer be perceived as an isolated declaration for change. If an organisation is to preserve the ability to adapt, a requirement for change must emerge from a continuum of such declarations joined by multiple dimensions of connectivity.

The simplifying assumptions that inform many approaches to requirements engineering apply when organisations are in a state of stasis or order. Key assumptions in this arena are that the domain expert has a complete understanding of a need for change and is capable of articulating fully the details of that need with precision and clarity. When the influence of chaos draws an organisation into the universe of complexity (and possibly beyond), the characteristics of the need for change become shrouded in complexity and uncertainty, i.e. the testimony of the domain expert can no longer be relied upon.

The challenge for the analyst is to devise an environment where the domain expert is enabled to give reliable accounts of organisational requirements. Development of such an environment, however, requires the integration of many interdependent techniques and artefacts. The obvious starting point is hard system methods, the body of techniques used by the analyst to elicit testimony from the domain expert.

### **8.3 The limitations of hard system methods**

Organisations are embroiled increasingly in a universe of complexity. To survive in this hostile environment, an organisation has to navigate a precarious trajectory between the competing forces of order and chaos; to succumb to either is to risk collapse from either stagnation or disintegration respectively.

Few would argue that hard system methods have, at the very least, made some contribution to describing an orderly environment. However, organisations are now drawn inexorably towards a chaotic environment

where every choice, decision and action is subjected to unprecedented complexity and uncertainty. When confronted by such challenges, the theoretic edifice for many hard system methods is beginning to crumble with the unravelling of many simplifying assumptions. Accordingly, the influence of hard system methods has diminished to the point where it is seen as little more than a marginal activity.

Survival in the universe of complexity requires that the organisation negotiate the competing forces of order and chaos. Negotiation requires of the organisation that it has the ability to adapt, i.e. the organisation must behave as a complex adaptive system. Adaptation sets a new agenda for hard system methods. Without the unifying cognitive influence of hard system methods, adaptation is unattainable. Unfortunately, the current scope and efficacy of hard system methods do not provide a credible basis for corporate adaptation.

The pioneering hard system methods were constrained by metaphors that were too primitive and, in some cases, conceptually flawed. The later OO methods addressed many of the earlier problems but have proved not to be scaleable to the corporate dimension. Without doubt, hard system methods still have much to offer the analyst. Within the universe of complexity, however, the contention is that a new approach to hard system methods is necessary if they are to guide the adaptation of the organisation.

#### **8.4 The role of cognitive psychology**

Cognitive psychology offers a possible explanation for the limited influence of hard system methods. The use of metaphors is a key cognitive weapon in the struggle to manage complexity. Metaphors control what is revealed and what is concealed in the bid to overcome complexity. Metaphors are used extensively by hard system methods to represent aspects of complex and uncertain concepts. If the metaphors are primitive or anomalous, they will soon be placed under strain as the analysis approaches the realms of complexity. Unfortunately, metaphors form a fundamental element of cognitive strategies and if they are

demonstrated to be placed easily under strain, the resulting cognitive strategy is likely to be abandoned by domain experts as they strive to create a mental model of a domain..

Great care must, therefore, be taken with the definition of metaphors that are destined to form the conceptual basis for a hard system method. Cognitive theory suggests that metaphors consist of two essential dimensions: abstraction and codification. Both dimensions support cognitive strategies that are highly cohesive and complementary.

The challenge for the analyst is to devise a metaphorical framework derived from abstraction and codification conventions that have the capacity to support sustained inquiry into a domain. Such a challenge is decidedly non-trivial and points to the need for a pluralistic approach to analysis.

## **8.5 Towards a sustainable methodological plurality**

Although popular methodologists may claim otherwise, there is no single method that offers the analyst all the necessary techniques to fully investigate a domain. The analyst is forced, inevitably, to adopt a methodological pluralism. As indicated earlier, however, many methods present cognitive barriers because they are built on metaphors that are under strain. The analyst may, therefore, certainly expect formidable cognitive barriers to be erected when seeking to integrate techniques from disparate methodological paradigms.

The thesis proposes that those concepts common to the selected paradigms must be reinvented such that they share a conceptual unity. This is not to suggest that the purpose of a concept is, in any sense, subverted. In this context, conceptual reinvention constrains the concept to further reinforce purpose. In fact, many of the concepts available with less sophisticated methods offer too much room for interpretation and, consequently, lose essential utility.

Moreover, it is contended that analysis must be grounded in the behaviour of a domain. Cognitive theory suggests that it is through behaviour that an expert first apprehends the essence of a domain and interprets other facets of a domain thereafter. It may be argued, therefore, that any attempt at methodological pluralism must be based on a common core of concepts that describe behaviour. Other features of the domain, e.g. business objects, organisational and technological topology, objectives and goals, constraints, culture, values and beliefs may be considered separately and associated with configurations of behavioural nodes through some form of complex mapping. It is through the mappings that the impact of shifts in behaviour may be traced to instances of non-behavioural nodes.

Before a pluralistic approach may be applied to analysis, however, it is necessary to develop essential patterns of analysis that constrain the behavioural core and its associations with non-core elements.

## **8.6 Patterns of analysis**

The thesis presents a set of generic patterns that provide a conceptual foundation for methodological pluralism.

The most fundamental of these patterns is the Behaviour Pattern that specifies characteristics that must be discernible for all behaviour nodes at all levels of abstraction, i.e. from the entire corporation to an atomic action. By far the most important characteristic is 'singularity of purpose' which states quite simply that a behavioural node may only have one purpose and, conversely, that a purpose can only be fulfilled by one behavioural node; i.e. purpose cannot be shared. The Behaviour Pattern also addresses the intimate relationship between behavioural nodes and events.



The next generic pattern addressing aspects of organisational behaviour is the Behaviour Typology Pattern that allows the analyst to specify the types of behaviour under scrutiny and the associations between those types. All types of behaviour must conform to the fundamental Behaviour Pattern.

The Business Object Pattern describes how business objects and their associations are derived from the transformation activities of behavioural nodes. In accordance with OO theory, it is assumed that the objects collaborate in some complex way to achieve the behaviour required of the domain.

Finally, the Domain Synthesis Pattern seeks to establish associations between the various nodes populating the different views of the domain.

While the Behaviour Pattern is considered to be inviolable, the other patterns can be modified at the discretion of the analyst to address the particular challenges of the domain of concern. It is expected that such modifications will conform to some pluralistic methodological convention.

These generic patterns provide a conceptual foundation for populating a domain with a configuration of nodes that describe some feature of the domain. The configurations also yield patterns that are described variously as generative patterns and domain patterns, and form the focus for much of the literature on analysis patterns. The case study provides examples of generative and domain patterns found in a Investment Banking domain.

From the perspective of cognitive theory, generic patterns form the metaphorical foundation for the methodology. The full descriptions of each pattern seek to establish the abstraction and codification conventions. Less attention is paid to the codification convention, as this will usually be adapted to address the complexities of the domain.

## **8.7 The essential features of a corporate knowledge base**

It has been argued in the earlier paragraphs that as organisations venture deeper into the realms of complexity, they are encountering problems for which hard system methods have little to offer. The domain expert is being placed in a position of cognitive isolation, and it is from this position that the expert is expected increasingly to pronounce on requirements for change.

It is also argued, however, that a judicious blend of methodological pluralism and 'behaviour accented' generic patterns better equip the analyst and domain expert to approach a complex domain. However, to leave matters here is to ignore the essential contribution of the corporate knowledge base to the analysis process. The thesis presents a description of the essential features characterising a corporate knowledge base. It is proposed that the corporate knowledge base is composed of a series of domains representing some feature of the organisation. A complex web of rich mappings associates nodes within and across the domains. It is worth noting here that a domain is considered to be a flexible construct that may be used to describe an entire system or some subset or view of the system. Also, within each domain, nodes of the same type may exist at various levels of abstraction. It should come as no surprise that the domain describing the organisational behaviour resides at the conceptual core of the corporate knowledge base and all other domains are subordinate to this core.

The corporate knowledge base is populated by knowledge assets reflecting the fruits of cognitive activity by the domain expert; it is to be expected that a knowledge asset will impact multiple domains and thus invoke enrichment of other assets. The opportunities for such enrichment, whether serendipitous or otherwise, will be greatly improved if the knowledge asset configurations for each domain exploit any available generative and domain patterns.

It is argued that as the corporate knowledge base matures in accordance with these constraints, it provides a potent reservoir of knowledge assets to stimulate speculative and formative reflection by the domain expert.

## **8.8 A new paradigmatic convention**

Knowledge sharing is an essential feature of the knowledge creation process; unless knowledge is shared between domain experts, the prospects for creating new knowledge assets are severely diminished. Moreover, the imperative to create knowledge assets is ever present as a complex adaptive system ensures that the scarcity and utility, i.e. the economic value, of existing knowledge is constantly under assault.

Cognitive theory suggests that knowledge can be created anywhere on a spectrum delineated by explicit and tacit knowledge, and that navigation of the spectrum is described by a spiral trajectory. Explicit knowledge describes the state where the abstraction and codification descriptors are fully evolved, and the knowledge asset is effectively in the public domain. Conversely, tacit knowledge describes the state where the abstraction and codification descriptors are, at best, embryonic and the knowledge asset is buried deep within the subconscious of the domain expert. Knowledge is considered to be created when there is significant movement towards either of these polar extremities.

For a corporate knowledge base to be effective, it must provide a conceptual foundation for knowledge creation. To achieve this it must reflect the cognitive priorities and strategies of the domain expert, i.e. the knowledge base must accentuate behaviour and exploit generative and domain patterns. Within these design constraints, new knowledge is created increasingly from the reservoir of existing knowledge assets as the knowledge base matures. This, of course, is a demonstration of the adaptability of the knowledge base in the face of shifting circumstances.

The domain expert is no longer stranded in a state of cognitive isolation. When formulating a requirement for change, the domain expert can use the corporate knowledge base to identify the requirement and specify a definition that is commensurate with the quality of knowledge assets. Thereafter, the corporate knowledge base is enriched further with details of the change requirement.

This scenario provides a new paradigm from which to reconstruct hard system methods.

## **8.9 Analysis – an unending quest for knowledge**

The implications of the complex adaptive system extend beyond reconstruction of hard system methods; they also challenge the purpose and conduct of analysis.

It can be argued that, in accordance with conventional wisdom, analysis is a process of externalisation where the domain expert is enabled to explicate tacit knowledge and record the outcome in the corporate knowledge base. The analysis process should also enable the internalisation process, i.e. enable tacit reflection of knowledge held in the public domain. The erosion of the value of knowledge has elevated the status of internalisation to that of a key knowledge creation process.

The analyst is then tasked with presenting explicit knowledge in such a form as that it eases the passage of internalisation for the domain expert. The analysis process is not complete with the production of some abstract representation of a statement of requirements; there is the need to prepare domain expert(s) for the next iteration of the knowledge creation spiral. In this sense, the analysis process can be viewed as an unending quest for knowledge.

## 8.10 Opportunities for further research

A trace of positivism is detectable in all threads of hard system methods; indeed, the more naïve the method the greater the quotient of positivism. In a sense, this is understandable because hard system methods are afforded the luxury of a certain objectivity when defining system architectures; while a correct architecture may be an elusive concept, there is usually no doubt when an architecture is wrong (the consequences are there for all to see). However, as environmental turbulence grows, there is a corresponding tendency for 'correct' architectures to be an increasingly remote and transient attainment. This phenomenon is contriving to undermine confidence in the presumed positivism.

According to Mingers (1997) the combined assault of Heisenberg's 'uncertainty principle' and the work of Hanson (1958), Kuhn (1970) and Popper (1972) have dealt a similar blow to the natural sciences in the twentieth century. Not surprisingly, the field of management science was also a victim of a similar assault. Mingers makes the point that all in disciplines 'the acceptance of paradigm isolation and incommensurability began to break down and, in the last decade, the debate has turned to various forms of pluralism, in both, methodological and philosophical terms.' It is not clear that hard system methods are included in this group of disciplines.

Mingers claims that some form of methodological pluralism is highly desirable and endorses the proposal of Landry and Banville that pluralism should be used extensively in the development of information systems (Landry & Banville, 1992). Mingers cautions however that the term 'methodological pluralism' can be interpreted in many ways:

- Loose pluralism offers a variety of paradigms but provide little guidance on their application.
- Complementarism considers internally consistent paradigms to be more or less appropriate for particular situations (Jackson, 1991)

- Strong pluralism argues that 'most if not all' intervention situations would benefit from a blend of methodologies from different paradigms.

Three arguments are offered in support of strong pluralism. Real-world situations are highly complex and multi-dimensional, therefore different paradigms focussing on various aspects of the reality are needed to manage the complexity. It is rare for a paradigm to remain consistently effective throughout all phases of a project; therefore a variety of paradigms should be selected according to their particular strengths. Finally, consideration of 'methodological pluralism' is due, as general disenchantment with individual methods is forcing practitioners to combine methods without the benefit of a theoretical foundation.

The final point appears to have anticipated a trend in the hard system methods field where the Unified Modelling Language and The Open Method both represent the combination of earlier methods; the motivation could, of course, be nothing more than commercial expediency (hard system methods represent a very crowded marketplace).

A clear inference to be drawn from the thesis is that if hard system methods are to make any further contribution to our understanding of organisations as complex adaptive systems, the need for a paradigmatic pluralism is undeniable. The attempts of the Unified Modelling Language and The Open Method appear, however, to be too modest. It is argued that the theoretical foundation for a pluralistic approach to hard system methods should embrace concepts from cognitive psychology, knowledge management and organisational behaviour. This thesis is simply a tentative step into this area.

Finally, as Jackson reports, there is an established body of research dedicated to integrating the soft and hard approaches to information systems development (Jackson, 1997). The Multiview framework and methodology of Avison and Wood-Harper probably represents the earliest attempts at this fusion of analysis methods (Avison & Wood-Harper, 1990). By seeking to fuse Mumford's ETHICS and Checkland's SSM with hard methodologies, Multiview presents Jackson with the problem of identifying

the theoretical foundation from which the fusion of 'apparently contradictory' approaches is based. The response declared a preference for moving from the softer to the harder methods (Wood-Harper, 1985). This declaration provoked the comment that 'either the "soft" rationalities must be distorted by the expectation that they will lead to a more structured intervention or the "hard" rationalities will suffer because they are operating in a hermeneutic climate and are front-ended by a soft logic'. A more recent response declared that 'people close the theory in action' by adapting Multiview to the complexities and uniqueness of each situation it encounters (Wood-Harper & Avison, 1992; Watson & Wood-Harper, 1995). Jackson cautions that as Multiview is now apparently non-prescriptive, it has acquired enhanced flexibility and adaptability, but is also vulnerable to pragmatism. Jackson draws the inescapable conclusion that plurality of rationalities must be based on a sound theoretical foundation.

The thesis identifies elements essential to a pluralistic rationality for hard system methods. A further area of research might be to extend the theoretical foundations suggested in the thesis for hard system methods such that they provide a focus for ETHICS and SSM debate, i.e. the fusion is directed from the harder to the softer methods.

### **8.11 The Contribution**

The discovery of a lucid theory comprising simple prescriptions presents an enticing proposition to the practitioner. This is particularly true of the analyst required to make sense of a domain exhibiting confusion, contradiction and paradox. Experiential reflection, however, has failed to reveal even the prospect of such a theory of analysis and has brought into question the validity of such a pursuit.

A unified theory of analysis arouses disquiet as it implies that the analyst can apply a set of simple prescriptions to a complex domain in some dispassionate and detached manner to produce a detailed account of a domain. Indeed, this is precisely the impression that popular methodologists wish to create

for their particular set of prescriptions. This approach is heavily influential in the formulation of current approaches to analysis. The fact that this approach is untenable in the universe of complexity provides a clue to the decline of hard system methods and analysis as a core system development activity.

In the universe of complexity the analyst can no longer be remote from the analysis process. It can be argued, that to simply apply these prescriptions in the hope that the complexities and uncertainties can be ignored (or will be resolved mysteriously) is simply to 'go through the motions' of analysis. The analyst must now become immersed in the analysis process. Before what is commonly accepted as analysis can commence, the analyst must interpret the challenges presented by the domain and the cognitive barriers likely to inhibit the domain expert in pursuing knowledge creation. Indeed, the cognitive predicament of the domain expert should be the dominant consideration in influencing the approach to analysis. Challenges presented by the domain and the cognitive profile of the domain expert is likely to change character during the analysis process; this should be reflected by shifts in the corresponding approach to analysis.

The thesis has sought to establish that the analysis process is both complex and uncertain, and is misrepresented somewhat by methods and techniques offering simple prescriptions. Experience from numerous assignments as an independent consultant has revealed an approach to analysis that builds on conventional prescriptions. This approach has indicated that rather than the twin characteristics of complexity and uncertainty presenting a barrier to analysis, they create unexpected opportunities for profound discoveries and inventions when absorbed into the analysis process.

The thesis represents exploratory research into the nature of analysis and has made a series of claims for revising the conduct and focus of analysis. Subjecting these claims to the rigour of formal verification provides a good opportunity for further research. Of particular interest here is establishing the limitations of the claims.



With the benefit of formal verification, it may be possible to consolidate the claims of the thesis into a cohesive and coherent theoretical foundation.

## **9 Bibliography**

1. Ackoff, R. (1970). *A Concept of Corporate Planning*. John Wiley & Sons
2. Aldrich, H.E. (1979). *Organisations and Environments*. Englewood Cliffs, NJ. Prentice-Hall
3. Alexander, C. (1979). *The Timeless Way of Building*. Oxford University Press
4. Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., Angel, S. (1977). *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press
5. Ansoff, H. (1975). *Managing Strategic Surprise by Response to Weak Signals*. California Management Review
6. Ansoff, I., McDonnell, E. (1990). *Implanting Strategic Management*. Prentice-Hall
7. Argenti, J. (1980). *Practical Corporate Planning*. Allen & Unwin
8. Argyris, C. (1977). Double Loop Learning in Organisations. *Harvard Business Review*. September-October
9. Argyris, C. (1982). *Reasoning, Learning and Action*. Jossey-Bass
10. Argyris, C. (1990). *Overcoming Organisational Defences: Facilitating Organisational Learning*. Addison-Wesley
11. Argyris, C., Putman, R., McLain Smith, D. (1985). *Action Science*. Jossey-Bass
12. Argyris, C., Schön, D. (1974). *Theory in Practice*. Jossey-Bass
13. Argyris, C., Schön, D. (1978). *Organisational Learning: A Theory of Action in Perspective*. Addison-Wesley
14. Ashby, W. (1956). *An Introduction to Cybernetics*. Chapman & Hall
15. Avison, D., Wood-Harper, T. (1990). *Multiview*. Blackwell Scientific Publications, Oxford, UK
16. Barker R. (1989). *CASE\*Method™: Entity Relationship Modelling*. Addison-Wesley
17. Bateson, G. (1979). *Mind and Nature. A Necessary Unity*. New York. Bantam Books
18. Batttram. A. (1996). *Managing Complexity*. The Industrial Society.

19. Bear, S., Allen, P., Coleman, D., Hayes, F. (1990). Graphical Specification of Object-Oriented System. Object-Oriented Programming Systems, Languages and Applications. Ottawa, Canada. OOPSLA'90.
20. Beck, K., Cunningham, W. (1989). A Laboratory for Teaching Object-Oriented Thinking. SIGPLAN Notices. Vol. 24(10)
21. Beer, S. (1985). Diagnosing the System for Organisations. John Wiley & Sons
22. Bellin, D., Simone, S. S. (1997). The CRC Card Book. Addison-Wesley Object-Oriented Software Engineering Series
23. Bertalanffy, L. von (1968). General Systems Theory. George Braziller. New York
24. Blauner, R. (1964). Alienation and Freedom. Chicago. Chicago University Press
25. Boehm, B.W., (1986). A spiral model for software development and enhancement. Software Engineering Notes. 11(4). ACM SIGSOFT Software Engineering Notes
26. Böhm C., Jacopini, G. (1966) 'Flow Diagrams, Turing Machines, and languages with Only Two Formation Rules. Communications from ACM. May. pp 366-371. Reprinted in Yourdon (1979).
27. Boisot, M. H. (1995). Information Space: A framework for learning in organisations, institutions and culture. Routledge.
28. Booch, G. (1991). Object-Oriented Design with Applications. Benjamin/Cummings
29. Booch, G. (1994). Object-Oriented Analysis and Design with Applications. 2nd Edition. Benjamin/Cummings
30. Booch, G., Rumbaugh, J., Jacobson, I. (1998). Unified Modeling Language User Guide. Object Technology Series. Addison-Wesley.
31. Borgida, A. T. (1984). Generalisation/Specialisation as a Basis for Software Specification. In On Conceptual Modelling (Ed. M. I. Brodie *et al.* Springer-Verlag. New York
32. Boston Consulting Group. (1972). Perspective On Experience. Boston
33. Bower, J. L. (1973). 'Simple Economic Tools For Strategic Analysis.' Harvard Business School Case Study. No. 9-373-094
34. Brooks, F. (1975). The Mythical Man-Month: Essays on Software Engineering. Addison-Wesley

35. Brooks, F. (1987). "No Silver Bullet". IEEE Computer. April
36. Buaron, . (1981). 'New Game Strategies.' The McKinsey Quarterly. (Spring 1981). pp. 22-40
37. Burns , T., Stalker, G. (1961). The Management of Innovation. Tavistock Publications
38. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M. (1996). A System Of Patterns: Pattern-Oriented Software Architecture. John Wiley & Sons
39. Capra, F., Steindl-Rast, D., Madison, T. (1991). Belonging to the Universe. San Francisco. Harper
40. Charan,R. (1991). How networks reshape organisations - for results. Harvard Business Review. Vol. 12. (Sept - Oct). pp 479 - 494
41. Checkland, P. (1981). Systems Thinking, Systems Practice. John Wiley & Sons
42. Checkland, P., Scholes, P. (1990). Soft Systems Methodology In Action. John Wiley & Sons
43. Chen, P. (1976). 'The entity relationship model - towards a unified view of data'. ACM Trans. Database Systems. 1(1), 9-36
44. Child, J. (1984). Organisation. Harper & Row
45. Claxton, G. (1997). Hare Brain, Tortoise Mind. London. Forth Estate
46. Coad, P. (1989). OOA: Object-Oriented Analysis. American Programmer.
47. Coad, P. North, D. Mayfield, M. (1995). Object Models: Strategies, Patterns, & Applications. Yourdon Press Computing Series
48. Coad, P., Yourdon, E. (1990). Object-Oriented Analysis. Prentice-Hall International
49. Coad, P., Yourdon, E. (1991). Object-Oriented Analysis. 2d Edition. Prentice-Hall International
50. Codd, E. F., (1970). 'A relational model of data for large shared data banks' Comm. ACM, 13, 377-87
51. Cohen, B. (1997). The Edge Of Chaos: Financial Booms, Bubbles, Crashes and Chaos. John Wiley
52. Cohen, J., Stewart, I. (1994). The Collapse of Chaos: Discovering Simplicity in a Complex World. Penguin Books
53. Coleman, D., Arnold, P., Bodoff, S., Gilchrist, H., Hayes, F., Jeremes, P. (1994). Object-Oriented Development: The Fusion Method. Prentice Hall

54. Cook, S., Daniels, J. (1994). *Designing Object Systems: Object-Oriented Modeling with Syntropy*. Prentice-Hall.
55. Coplien, J. O., Schmidt, D. C., (1995). *Pattern Languages of Program Design*. Addison-Wesley.
56. Dahl, O., Dijkstra, E., Hoare, C. A. R. (1972). *Structured Programming*. London, England: Academic Press.
57. Darnton, G., Darnton, M. (1997). *Business Process Analysis*. Thomson Business Press
58. Date, C. J. (1983). *An Introduction to Database Systems*. Reading, Mass. Addison-Wesley
59. Davenport, T. H. (1993). *Process Innovation: Reengineering Work through Information Technology*. Harvard Business School Press
60. Davenport, T., Short, J. (1990). The new industrial engineering: information technology and business process redesign. *Sloan Management Review*, Summer.
61. DeGrace, P., Hulet Stahl, L. (1993). *The Olduvai Imperative: Case and the State of Software Engineering Practice*. Prentice-Hall.
62. DeMarco, T. (1978). *Structured Analysis and Design Specification*. New York. Yourdon Press.
63. DeMarco, T. (1979). *Structured Analysis and Systems Specification*. New York. Yourdon Press
64. Donellon, , A., Gray, B., Bourgon, M. (1986). *Communication, Meaning and Action*. Administrative Science Quarterly
65. Downs, E., Clare, P., Coe, I. (1988). *Structured systems analysis and design method: Application and Context*. 2<sup>nd</sup> ed. Prentice-Hall
66. Dreger, B. (1989). *Function Point Analysis*. Prentice-Hall
67. Edwards, J. (1984). *Process Architecture Design Technology*. Internal reports from Associative Design Technology. Westborough. Mass.
68. Eriksson, H-E., Penker, M. (1998). *UML Toolkit*. John Wiley & Sons
69. Espejo, R., Harden, R. (1989). *The Viable Systems Model: Interpretation and Application of Stafford Beer's VSM*. John Wiley & Sons
70. Espejo, R., Harden, R. (1989). *The Viable Systems Model: Interpretation and Application of Stafford Beer's VSM*. John Wiley & Sons

71. Eva, M. (1994). SSADM Version 4: A User's Guide. 2<sup>nd</sup> ed. McGraw-Hill
72. Festinger, L. (1957). The Theory of Cognitive Dissonance. Palo Alto, California. Stanford University Press
73. Festinger, L., Schachter, S., Back, K. (1950). Social pressures in informal groups: A study of a housing project. New York. HarperCollins
74. Flood, R., Romm, N. (1996). Diversity Management: Triple Loop Learning. John Wiley
75. Forrester, J. (1975). The Counter-Intuitive Behaviour of Social Systems, in Collective Papers of J.W. Forrester. Wright-Allen Press
76. Fowler, M. (1997). Analysis Patterns: Reusable Object Models. Addison-Wesley.
77. Fowler, M., Scott, K. (1997). UML Distilled: applying the Standard Object Modeling Language. Addison-Wesley.
78. Franke, R. (1987). Technological revolution and productivity decline: Computer introduction in the financial industry. Technology Forecasting and Social Change, Vol. 31.
79. Galliers, R. (1995). 'IT and Organisational Change: Where Does BPR Fit In?'. In Examining Business Process Re-engineering: Current Perspectives and Directives. Editors. (Burke, G & Peppard, J). The Cranfield Management Series. Kogan Press
80. Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional Computing Series.
81. Gane, C., Sarson, T. (1979). Structured System Analysis: Tools and Techniques. Prentice-Hall
82. Gell-Mann, M. (1994). The Quark and the Jaguar. W.H. Freeman
83. Gick, M., Holyoak, K. (1983). Schema introduction and analogical transfer. Cognitive Psychology. Vol. 15
84. Gilb, T. (1988). Principles of Software Engineering Management. Addison-Wesley
85. Gleick, J. (1988). Chaos: The Making of a New Science. Heinemann
86. Gluck, F. W. 'Strategic Resources and Resource Allocation.' The McKinsey Quarterly (Winter 1980). pp. 22-23

87. Goodman, M., Kemeny, J. (1994). Archetypes in The Fifth Discipline Fieldbook. (Edited by P. Senge) Nicholas Brealey Publishing
88. Graham, I. (1991). Object-Oriented Methods. Addison-Wesley
89. Graham, I. (1994). Object-Oriented Methods. 2nd Edition. Addison-Wesley
90. Graham, I., Henderson-Sellers, B., Younessi, H. (1997). The OPEN Process Specification. ACM Press. Addison-Wesley
91. Grant, R. (1991). 'The resource-based theory of competitive advantage: Implications for strategy formulation'. California Management Review. pp 114-135
92. Halé, J. (1995). From Concepts to Capabilities: Understanding and exploiting change as competitive advantage. John Wiley & Sons
93. Hall, A. III (1989). Metasystems Methodology. Pergamon Press. Oxford
94. Hamel, G., Prahalad, C. (1994). Competing for the Future. Harvard Business School Press
95. Hammer, M., Champy, J. (1993). Reengineering the Corporation: A Manifesto for Business Revolution. Nicholas Brealey
96. Hammer, M., Stanton, S. (1995). The Reengineering Revolution. HarperCollins
97. Hampden-Turner, C. (1990). Charting the Corporate Mind. New York. Free Press, Macmillan
98. Hannan, M., Freeman, J. (1977). The population ecology of organisations. American Journal of Sociology. Vol. 83. pp. 929-64
99. Hanson, N. (1958). Patterns of Scientific Discovery. Cambridge University Press
100. Harel, D. (1987). Statecharts: A visual formalism for complex systems. Science of Computer Programming, 8. pp 231-274
101. Hay, D. (1996). Data Model Patterns: Conventions Of Thought. Dorset House Publishing. New York
102. Hayes, R., Abernathy, W. (1980). Managing our way to economic decline. Jul-Aug. Harvard Business Review
103. Hedley, B. (1977). Strategy and the Business Portfolio. Long Range Planning Vol. 10(2). Boston Consulting Group

- 104.Henderson-Sellers, B. (1997). The OPEN Process: Activities and Tasks. Object Currents. SIGS Publications (<http://www.sigs.com/publications/docs/oc/9701/sellers.html>)
- 105.Hertha, W. (1995). The Architecture Reference Model. OOPSLA'95. SIGS Publications. (<http://www.tiac.net/users/jsuth/oopsla/herta.html>)
- 106.Holdsworth, J. (1994). Software Process Design: Out of the Tar Pit. McGraw-Hill
- 107.Holton, G. (1986). The Advancement of Science and its Burdens: The Jefferson Lectures and Other Essays. Cambridge: Cambridge University Press.
- 108.Humphrey, W. (1989). Managing the Software Process. Addison-Wesley.
- 109.Issacs, W. (1992). The Ladder Of Inference. MIT Centre for Organisational Learning
- 110.Jackson, M. (1991). Systems Methodology for Management Sciences. Plenum Press. New York
- 111.Jackson, M. (1997). Pluralism in Systems Thinking and Practice. In (Mingers, J., Gill, A.) Multi-Methodology: The Theory and Practice of Combining management Science Methodologies. John Wiley & Sons
- 112.Jackson, M. A. (1975). Principles of Program Design. London. Academic Press.
- 113.Jackson, M. A. (1983). System Development. Prentice-Hall
- 114.Jackson, M., Twaddle, G. (1997). Business process Implementation: Building Workflow Systems. Addison-Wesley
- 115.Jacobson, I. (1994a). Basic Use-Case Modeling. Report On Object Analysis and Design. 1(2). pp 15-19. SIGS Publications
- 116.Jacobson, I. (1994b). Basic Use-Case Modeling. Report On Object Analysis and Design. 1(3). pp 7-9. SIGS Publications
- 117.Jacobson, I. (1994c). Use Cases and Objects. Report On Object Analysis and Design. 1(4). pp 8-10. SIGS Publications
- 118.Jacobson, I. (1995). Use Cases in Large-Scale Systems. Report On Object Analysis and Design 1(6). pp. 9-12. SIGS Publications
- 119.Jacobson, I., Booch, G., Rumbaugh, J. (1998). The Objectory Software Development Process. Object Technology Series. Addison-Wesley.



120. Jacobson, I., Christerson, M., Jonsson, P., Övergaard, G. (1992). Object-Oriented Software Engineering: A Use Case Driven Approach. ACM Press. Addison-Wesley
121. Jacobson, I., Ericsson, M., Jacobson, A. (1995). The Object Advantage: Business Process Reengineering with Object Technology. ACM Press. Addison-Wesley
122. Jacobson, I., Griss, M., Jonnson, P. (1997). Software Reuse: Architecture, Process and Organisation for Business Success. Addison-Wesley
123. Jayaratna, N. (1988). Guide to Methodology Understanding and Information Systems Practice. International Journal of Information Management. Vol 8, No. 1
124. Jayaratna, N. (1994). Understanding and Evaluating Methodologies: NIMSAD, A Systemic Framework. McGraw-Hill
125. Johansson, H., McHugh, P. Pendlebury, A., Wheeler, W. III. (1993) Business Process Reengineering: Breakpoint Strategies for Market Dominance. John Wiley & Sons
126. Johnson, G., Scholes, K. (1989). Exploring Corporate Strategy. Prentice-Hall
127. Johnson-Laird, P. (1988). The Computer and the Mind. Harvard University Press
128. Jones, C. B., (1990), Systematic Software Development using VDM, 2nd Ed. Prentice Hall International, Englewood Cliffs, NJ, 1990
129. Jordon, N. (1968). Themes in Speculative Psychology. London: Tavistock
130. Kemeny, J., Goodman, M, Karash, R. (1994). Starting With Storytelling (Systems Thinking) in The Fifth Discipline Fieldbook. (Edited by P. Senge) Nicholas Brealey Publishing
131. Kent, W. (1990). A Framework for Object Concepts. Hewlett-Packard. Report HPL-90-30.
132. Kets De Vries, M. (1980). Organisational Paradoxes: Clinical approaches to Management. London. Tavistock Publications
133. Kim, D. (1993). Seven Steps for Breaking Through Organisational Gridlock. The Systems Thinker
134. Kitson, D., Masters, S. (1993). *An Analysis of SEI Software Process Assessment Results, 1987-1991*. In Proceedings of the Fifteenth International Conference on Software Engineering. 68-77. Washington, D.C. IEEE Computer Society Press
135. Klir, G. J. (1991). Facets of Systems Engineering. Plenum. New York

136. Kolb, D. (1976). *The Learning Style Inventory: Technical Manual*. Boston, Mass., McBer and Co
137. Kosko, B. (1992). *Neural networks and Fuzzy Systems*. Englewood Cliffs, NJ: Prentice-Hall.
138. Kuhn, T. (1962). *The Structure of Scientific Revolution*. University Of Chicago Press
139. Lakoff, G. (1987). *Women, Fire and Dangerous Things. What Categories Reveal About The Mind*. Chicago, IL: The University Of Chicago Press
140. Lakoff, G., Johnson, G. (1980). *Metaphors We Live By*. University Of Chicago Press
141. Landauer, T. (1995). *The Trouble With Computers. Usefulness, Usability, and Productivity*. Massachusetts Institute of Technology
142. Landry, M., Banville, C. (1992). A disciplined methodological pluralism for MIS research. *Accounting, Management & Information Technology*. Vol. 2(2). pp. 77
143. Langer, S.K. (1967). *An Introduction To Symbolic Logic*. (3<sup>rd</sup> Edition). Dover Publications. New York
144. Lawrence, P., Lorsch, J. (1967). *Organisation and Environment*. Harvard University Press.
145. Leavitt, H. (1965). 'Applying Organisational change in Industry: Structural, Technological and Humanistic Approaches. In *Handbook of Organisations*. Editors (J. March). Rand McNally. Chicago
146. Leavitt, H. (1975). *Beyond the Analytic Manager*. California Management Review
147. Leonard-Barton, D. (1992). Core Capabilities and Core Rigidities: A Paradox in Managing New Product Development. *Strategic Management Journal*, 13: pp 111-125
148. Lewicki, P., Hill, t., Czyzewska, M. (1992). Nonconscious acquisition of information. *American Psychologist*. Vol. 47
149. Lewin, K. (1947). Feedback problems in social diagnosis. Part II-B of *Frontiers in Group Dynamics*. Human Relations. Vol 1. pp. 147-153
150. Lovelock, J. (1991). *Healing Gaia: Practical Medicine for the Planet*. Harmony Books
151. Loveman, G. (1986). *The productivity of information technology capital: An economic analysis*. MIT
152. Loveman, G. (1990). *An assessment of the Productivity Impact of Information Technologies*. Cambridge, MA: MIT. Sloan School of Management

153. MacDonald, I., Palmer, I. (1982). System development in a shared data environment, the D2S2 methodology. In Olle *et al.* (1982)
154. Makridakis, S. (1990). Forecasting, Planning and Strategy for the 21<sup>st</sup> Century. New York. Free Press
155. Mandelbrot, B. (1998). Fractals and scaling in finance: Discontinuity, concentration and risk. Springer
156. Martin, J. (1986). Information Engineering. Volume 2: Strategies and Analysis. Savant Research Studies
157. Martin, J. (1987a). Information Strategy Planning Handbook. James Martin Associates
158. Martin, J. (1987b). Business Area Analysis Handbook. James Martin Associates
159. Martin, J. (1989). Information Engineering. Prentice Hall
160. Martin, J. (1996). Cybercorp: the new business revolution. Amacom. American Management Association
161. Martin, J., Finkelstein, C. (1981). Information Engineering, Vols 1 and 2. Prentice Hall
162. Martin, J., Odell, J. (1992). Object-Oriented Analysis and Design. Prentice-Hall.
163. Martin, J., Odell, J. (1995). Object-Oriented Methods: A Foundation. Prentice-Hall
164. Martin, J., Odell, J. (1996). Object-Oriented Methods. Pragmatic Considerations.. Prentice-Hall
165. McConnell, S. (1996). Rapid Development. Taming Wild Software Schedules. Microsoft Press.
166. McKelvey, W. (1981). Organisational Systematics. Los Angeles. University of California Press
167. McMenamin, S., Palmer, J. (1984). Essential Systems Analysis. Yourdon Press
168. Meyer, B. (1988). Object-Oriented Software Construction. Prentice-Hall
169. Miller, D. (1986). Configurations of strategy and culture: towards a synthesis. Strategic Management Journal. Vol. 7. pp 233-49
170. Miller, E., Rice, A. (1967). System of Organisation: The control of tasks and sentient boundaries. London: Tavistock
171. Mingers, J. (1997). Towards Critical Pluralism. In (Eds Mingers, J., Gill, A.) Multi-Methodology. (1997). John Wiley

172. Mingers, J., Gill, A. (1997). *Multi-Methology: The Theory and Practice of Combining Management Science Methodologies*. John Wiley
173. Mintzberg, H. (1989). *Mintzberg on Management: Inside Our Strange World of Organisations*. Free Press. New York
174. Mintzberg, H. (1994). *The Rise and Fall of Strategic Planning*. Prentice-Hall
175. Mintzberg, H. (1996). *Beyond Configuration*. In Eds. Mintzberg, H., Quinn, J. *The Strategy Process: Concepts, Contexts, Cases*. Third Edition. Prentice-Hall
176. Mintzberg, H., McHugh, A. (1985). *Strategy Formation in an Adhocracy*. *Administrative Science Quarterly*
177. Mintzberg, H., Waters, J. (1985). *Of strategies deliberate and emergent*. *Strategic Management Journal*. Vol 6
178. Mueller, R. (1986). *Corporate networking: Building channels for information and influence*. New York: Free Press
179. Mumford, E. (1983a). *Designing Participatively*. Manchester Business School. Manchester
180. Mumford, E. (1983b). *Designing Human systems*. Manchester Business School. Manchester
181. Mumford, E. (1995). *Effective Systems Design and Requirements Analysis*. Macmillan Press
182. Mumford, E. (1996). *Systems Design: Ethical Tools for Ethical Change*. Macmillan Press
183. NCC. (1995). *SSADM 4+; Version 4.2, Vols 1 and 2*. NCC Blackwell, Oxford
184. Nevis, E., DiBella, A., Gould, J. (1997). *Understanding organisations as learning systems*, in *Core Competency-Based Strategy* (Editors: Campbell, A., Summers Luchs, K). Thomson Business Press
185. Nisbet, R. (1969). *Social Change and History. Aspects of the Western Theory of Development*. Oxford University Press
186. Nohria, N., Eccles, R. (1992). *Networks and Organisations*. Cambridge, MA. Harvard University Press
187. Nonaka, I., Takeuchi, H. (1995). *The Knowledge-Creating Company*. Oxford University Press
188. Norman, D. (1983). *Some observations on mental models*, in *Mental Models* (Editors Gentner, D. & Stevens, A). Lawrence Erlbaum Associates

- 189.Olle, T., Sol, H., Verrijn-Stuart, A. (1982). Information systems design methodologies; a comparative review. North Holland, Amsterdam
- 190.Ormerod, P. (1994). The Death of Economics. Faber and Faber.
- 191.Orr, K. (1981). Structured Requirements Definition. Ken Orr & Associates
- 192.Orr, K. (1977). Structured Systems Development. New York, NY: Yourdon Press
- 193.Ould, M. (1995). Business Processes: Modelling and Analysis for Re-engineering and Improvement. John Wiley & Sons
- 194.Page-Jones, M. (1988). The Practical Guide to Systems Design. Englewood Cliffs, NJ; Yourdon Press.
- 195.Pall, G. (1987). Quality Process Management. Prentice-Hall
- 196.Palmer, I., Rock-Evans, R. (1981). Data analysis. IPC Publications. Surrey
- 197.Pantin, C. (1968). The Relations Between the Sciences. London: Cambridge University Press
- 198.Parnas, D., Clements, P. (1986). A Rational Design Process. How and Why to Fake It. IEEE Transactions on Software Engineering. Vol. SE-12(2)
- 199.Partridge, C. (1996). Business Objects: Re-engineering for Reuse. Butterworth-Heinemann
- 200.Pascale, R. (1990). Managing on the Edge: How Successful Companies Use Conflict to Stay Ahead. Viking. Penguin
- 201.Peters, T. (1987). Thriving On Chaos. Macmillan
- 202.Peters, T. (1993). Liberation Management. Macmillan
- 203.Peters, T., Waterman, R. (1982). In Search Of Excellence. Harper & Row
- 204.Polanyi, M. (1958). Personal Knowledge: Towards a Post-Critical Philosophy. Routledge and Kegan Paul
- 205.Polanyi, M. (1966). The Tacit Dimension. Routledge & Kegan Paul
- 206.Popper, K. (1972). Objective Knowledge: An Evolutionary Approach. Oxford University Press. London
- 207.Popper, K. (1983). Realism and the Aim of Science. Hutchinson

208. Porter, M. (1980). *Competitive Strategy: Techniques for Analysing Industries and Competitors*. The Free Press
209. Porter, M. (1985). *Competitive advantage*. The Free Press. New York
210. Prahalad, C., Hamel, G. (1990). The Core Competencies of the Corporation. *Harvard Business Review*. May-June.
211. Pressman, R. (1992). *Software Engineering: A Practitioner's Approach*. McGraw-Hill
212. Prigogine, I., Stengers, I. (1984). *Order out of chaos: Man's new dialogue with nature*. Bantam Books
213. Putman, R. (1994a). 'Conversational Recipes' in *The Fifth Discipline Fieldbook*. (Edited by P. Senge) Nicholas Brealey Publishing
214. Putman, R. (1994b). 'Risks and opportunities with the left-hand column' in *The Fifth Discipline Fieldbook*. (Edited by P. Senge) Nicholas Brealey Publishing
215. Quinn, J. (1980). *Strategies for Change: Logical Incrementalism*. Irwin
216. Quinn, R., Cameron, K. (1988). *Paradox and transformation*. New York. Ballinger/HarperCollins
217. Reenskaug, T., Wold, P., Arild Lehne, O. (1996). *Working With Objects: The Ooram Software Engineering Method*. Manning Publications
218. Roach, S. (1985). *The New Technology Cycle*. New York. Morgan Stanley Economic Perspectives
219. Roach, S. (1992). *Technology Imperatives*. New York. Morgan Stanley.
220. Rockart, J. (1979). Chief executives define their own data needs. *Harvard Business Review*. March-April
221. Ross, R. (1994). 'Mental Models' in *The Fifth Discipline Fieldbook*. (Edited by P. Senge) Nicholas Brealey Publishing
222. Ross, R., Kleiner, A. (1994). 'The Left-Hand Column' in *The Fifth Discipline Fieldbook*. (Edited by P. Senge) Nicholas Brealey Publishing
223. Ross, R., Roberts, C. (1994). *Balancing Inquiry and Advocacy in The Fifth Discipline Fieldbook*. (Edited by P. Senge) Nicholas Brealey Publishing

224. Rowe, A., Mason, R., Dickel, K., Westcott, P., Snyder, N. (1989). Strategic Management and Business Policy: A Methodological Approach. Addison-Wesley
225. Rubin, K., Goldberg, A. (1992). Object Behaviour Analysis. Communications of the ACM. Vol. 35(9).
226. Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W. (1991). Object-Oriented Modeling and Design. Prentice-Hall
227. Rumbaugh, J., Booch, G., Jacobson, I. (1998). Unified Modeling Language Reference Manual. Object Technology Series. Addison-Wesley.
228. Schön, D. (1987). Educating the Reflective Practitioner: Toward a new design for teaching and learning in the professions. San Francisco. Jossey-Bass
229. Schön, D. A. (1991). The Reflective Practitioner: How Professionals Think in Action. New York. Basic Books
230. Scott Morton, M. (1991). The Corporation of the 1990s: Information Technology and Organisational Transformation. Oxford University Press
231. Senge, P. (1990). The Fifth Discipline: The Art and Practice of the Learning Organisation. Century Business
232. Senge, P. (1994). The Fifth Discipline Fieldbook: Strategies and Tools for Building a Learning Organisation. Nicholas Brealey Publishing
233. Shaw, (M. (1989). Larger Scale Systems Require Higher-level Abstractions. SIGSOFT Engineering Notes. Vol 14. (3).
234. Shlaer, S., Mellor, S. (1988). Object-Oriented Analysis: Modeling the World in Data. Yourdon Press. Prentice-Hall
235. Shlaer, S., Mellor, S. (1992). Object Lifecycles: Modeling the World in States. Yourdon Press. Prentice-Hall
236. Smith, F. (1983). Essays into literacy. Portsmouth. NH; Heinemann Press
237. Somerville, I. (1989). Software Engineering. Third Edition. Addison Wesley

238. Sommerville, I., Sawyer, P. (1997). Requirements Engineering. A Good Practice Guide. John Wiley & Sons
239. Spivey, M. (1990). The Z Notation: A Reference Manual. Prentice Hall International, Englewood Cliffs, NJ
240. Stacey, R. (1993). Strategic Dynamics and Organisational Behaviour. Pitman Publishing
241. Stacey, R. (1996). Complexity and Creativity in Organisations. San Francisco. Berret-Koehler Publishers
242. Stalk, G., Evans, P., Shalman, L. (1992). Competing on Capabilities: The New Rules Of Corporate Strategy. Harvard Business Review, March-April, pp 57-69
243. Stamper, R. (1988). 'Analysing the Cultural Impact of a System'. International Journal of Information Management. Vol. 8. (3)
244. Stepp, R., Michalski R. (1986). Conceptual Clustering of Structured Objects: A Goal-Oriented Approach. Artificial Intelligence. Vol. 28(1).
245. Strassmann, P. (1990). The Business Value of Computers: An Executive's Guide. New Canaan, CT. Information Economic Press.
246. Stroustrup, B. (1991). The C++ Programming Language. Second Edition. Addison-Wesley
247. Swieringa, J., Wierdsma, A. (1992). Becoming a Learning Organisation. Addison-Wesley
248. Tapscott, D. Caston, A. (1993). Paradigm Shift: The New Promise of Information Technology. McGraw-Hill.
249. Taylor, B., Sparkes, J. (1977). Corporate Strategy and Planning. Heinemann.
250. Taylor, D. (1992). Object-Oriented Information System: Planning and Implementation. John Wiley & Sons
251. Taylor, D. (1995). Business Engineering with Object Technology. John Wiley & Sons
252. Teece, D., Pisano, G., Shuen, A (1991). Dynamic Capabilities and Strategic Management. Centre for Research in Management. University of California. Berkeley



253. Teichroew, D., Macasovic, P., Hershey, E. A., Yamamoto, Y. (1980). *Application of the entity-relationship approach to information processing systems modelling* in Entity-Relationship Approach to Systems Analysis and Design, North-Holland, Amsterdam
254. Trist, E., Branforth, K. (1951). Some social and psychological consequences of a long-walled method of coal getting. *Human Relations*. Vol 5. pp 6-24
255. Tushman, M., Newman, W., Romanelli, E. (1986). Convergence and Upheaval: Managing the unsteady pace of organisational evolution. *California Management Review*. Autumn
256. Waldrop, M (1992). *Complexity: The emerging science at the edge of chaos*. Simon & Schuster
257. Warnier, J, D. (1974). *Logical Construction Of Programs*. Van Nostrand Reinhold
258. Warnier, J, D. (1981). *Logical Construction Of Systems*. Van Nostrand Reinhold
259. Waterman, R., Peters, T., Phillips, J. (1980). Structure is not organisation. June. *Business Horizons*.
260. Watson, H., Wood-Harper, T. (1995). Methodology as metaphor: the practical basis for multiview methodology (a reply to M.C. Jackson). *Information Systems Journal*. Vol. 5(3)
261. Weaver, P. (1993). *Practical SSADM Version 4; Complete Tutorial Guide*. Pitman
262. Weick, K. (1977). Organisational Design: organisations as self-organising systems. Autumn. *Organisational Dynamics*
263. Wheatley, M. (1993). *Leadership and the New Science*. San Francisco. Berrett-Koehler
264. Wirfs-Brock, R., Wilkerson, B., Weiner, L. (1990). *Designing Object-Oriented Software*. Prentice-Hall.
265. Wood-Harper, T. (1985). Research methods in information systems: using action research. In E. Mumford et al (Editors) *Research Methods in Information Systems*. Elsevier Amsterdam. pp. 169-191
266. Wood-Harper, T., Avison, D. (1992). Reflections from the experience of using Multiview: through the lens of soft systems methodology. *Systemist*, Vol 14. pp. 136-144
267. Woodward, J. (1965). *Industrial Organisation. Theory and Practice*. Oxford University Press
268. Yourdon, E. (1979). *Classics in Software Engineering* (ed. Ed Yourdon). Englewood Cliffs. NJ: Yourdon Press/Prentice Hall

269. Yourdon, E. (1989). *Modern Structured Analysis*. Prentice-Hall
270. Yourdon, E. (1992). *Decline and Fall of the American Programmer*. Yourdon Press. Prentice-Hall
271. Yourdon, E., Constantine, L. (1979). *Structured Design*. Englewood Cliffs, NJ; Prentice-Hall
272. Zuboff, S. (1988). *In the Age of the Smart Machine: The Future of Work and Power*. Heinemann

## **10 Original Sources**

1. Craik, K. (1943). *The nature Of Explanation*. Cambridge University Press
2. Ogden, C. K., Richards, I.A. (1923). *The Meaning Of Meaning*. (8th Edition., 1946). Harcourt, Brace, and World
3. Rogers, E., Shoemaker, F. (1971). *Communication in Innovation: A Cross Cultural Approach*. The Free Press
4. Russell, B. (1956). *Mathematical Logic as based on the Theory of Types*, reproduced in *Logic and Knowledge*. Unwin Hayman. London
5. Shannon, C., Weaver, W. (1949). *The Mathematical Theory Of Communication*. Urbana. The University of Illinois Press
6. Whitehead, A. N., Russell, B. (1910). *Principia Mathematica*. Cambridge University Press